1991年9月1日発行（毎月1回1日発行）第10巻9号通巻113号　昭和58年11月2日第三種郵便物認可

PERSONAL COMPUTER MAGAZINE for MZ, X1, and X68000

特集 MAGICをBrush up!

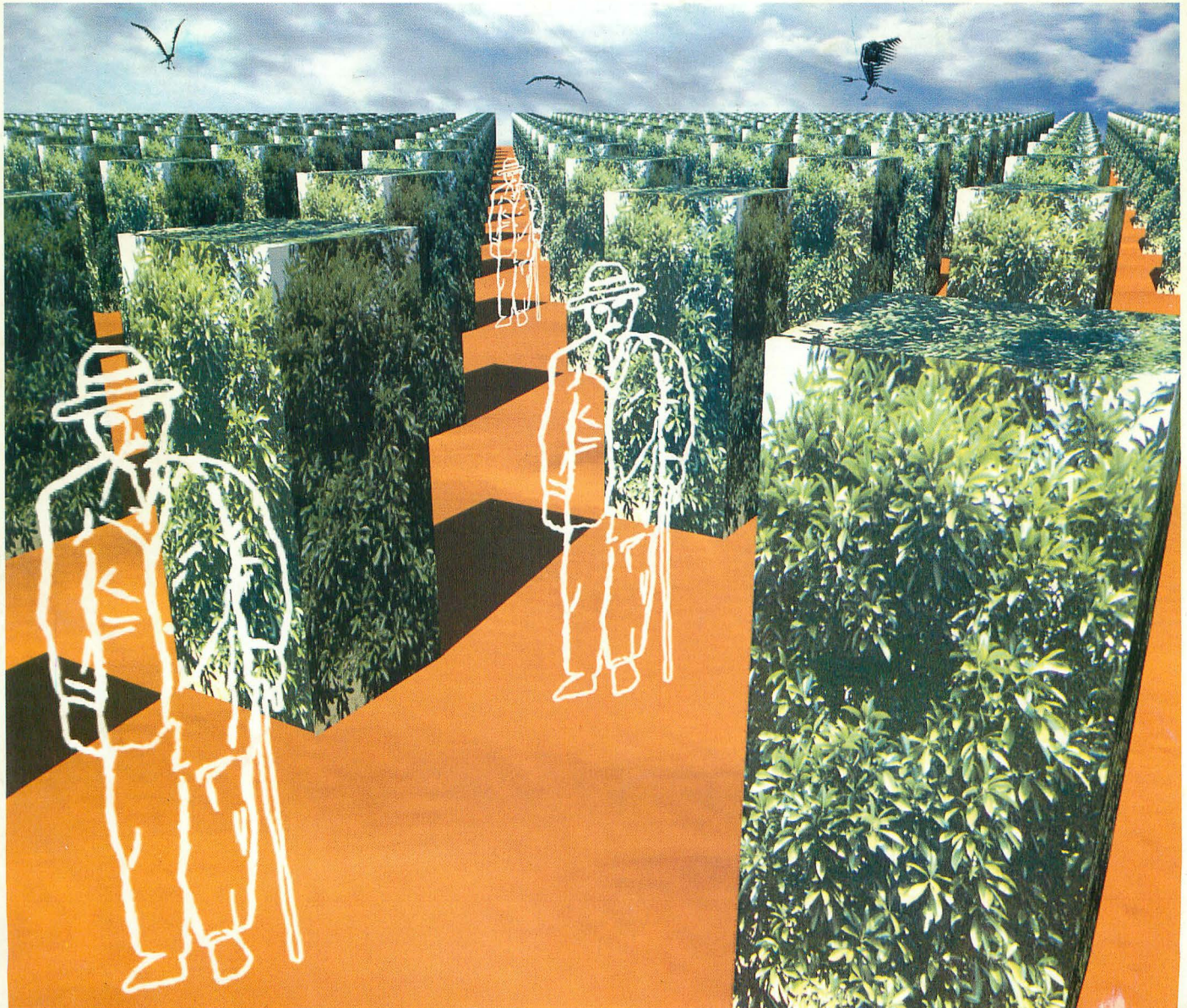
特別企画 Oh! Xの正しい読み方

X1用アスレチックアクションゲーム Manual Runner

大人のためのX68000「マルチワープロPRO-68K Multiword」

新製品 Mu-1 Super/NEW PrintShop PRO-68K ver.2.0

1991

SOFT BANK オー／エックス 定価600円

いまクロック16MHzの俊才、「エクシヴィ」のデビューで5年に及ぶ68000CPUへの探求は、ひとつの結論を得ようとしています。極めたといえば言い過ぎでしょうが、事の深淵に迫ろうと努力するもののみに与えられる深い充足を、私たちスタッフは、これまでX68000を支えていただいたユーザー、ソフトハウス、ハードベンダー諸兄とともに味わいたい心境です。徹底したこだわりと、それを裏付けるアドバンストテクノロジー、世間の逆風を揚力にしてしまう、それなりの魅力と知性を背景として備えたX68000が、パーソナルコンピュータに新しいジャンルを切り拓いてきた歩みは、ご存じの通りです。現在のマルチメディア環境を開発当初から想定していた先見性。一言でいえばクリエイティブマインドということでしょうが、そのグラフィックアビリティ、映像統合コンセプト、サンプリング音源、ウィンドウ環境、そうした単に、とはいえ凄いスペックさえ超えたところにX68000の付加価値は存在します。アプリケーションを走らせるだけのブラックボックス化した、あるいは文房具としてのマシン、それはそれで異論はないのですが、本来的にパーソナルコンピュータがもつ可能性を育む、いわば創造性という観点から物足りなさを覚えることも事実です。X68000は、ある意味ではたいへんな異端児かも知れません。しかし世間から見たその"異能"は、私たちが考えるパーソナルコンピュータとしてはまさにスタンダードに他なりません。いつも新鮮な感動がある、驚きがある。新しい発見がある。"センス"の違いはスペックをも超えて使う人に訴えかける、敢えて68000CPUに執着してきた理由もここにあります。ワークステーションとしての成熟、先見性、創造性の具現化、ユーザーインターフェイスの追求。X68000の進化の過程はここに凝縮されています。

——新しい「エクシヴィ」がこのコンセプトをどう発展させたか、ご体感ください。

瞬速16MHz、エクシヴィ快走。

16MHzクロック68000搭載：OSの高速化、アートワークをパワフルにサポートするクロック周波数16MHzの68000CPUを搭載。クリエイティブワークステーションにふさわしいシステムパフォーマンスを実現しました。

SX-WINDOW ver1.1搭載：CPUのクロックアップと合わせ、大幅な処理速度の向上を実現。操作性を一段と高めたニューバージョンです。多機能・高速の強力エディタを搭載。文字選択・外字作成ツールも装備して、スムーズな日本語入力環境をサポート。またプリンタデバイスドライバを搭載し、多彩な印字指定が可能です。もちろん、こうした新しい環境がすべてのX68000で享受できることは言うまでもありません。そして待望のウィンドウアプリケーションもリリースされはじめています。

高密度メモリ拡張環境：メインメモリは標準で2Mバイト、本体内部のメイン基板上に6Mバイト増設でき、I/Oスロットを使用せず最大8Mバイトの高速メモリアクセスを実現。さらにI/Oスロットへの増設を含め最大12Mバイトまで拡張できます。数値演算プロセッサも本体内に取り付けられます。

※2MB増設メモリ（ボード型）CZ-6BE2A　標準価格59,800円（税別）、2MB増設メモリ（チップ型）CZ-6BE2B　標準価格54,800円（税別）、数値演算プロセッサ（チップ型）CZ-6BP2　標準価格45,800円（税別）を使用。（すべて別売）

●大容量メディア対応、世界標準SCSIインターフェイス標準装備●X68000シリーズとフルコンパチブル設計●高品位なチタンブラックのニューデザインマンハッタンシェイプ●81MバイトSCSI仕様HDD搭載（CZ-644C）/内蔵可能（CZ-634C）●1024×1024ドットの実画面エリアを装備した高解像度表示（最大表示エリア768×512ドット・65,536色中16色表示）、65,536色同時表示（512×512ドット時）、先駆の高解像度自然色グラフィック●AD PCM、ステレオ8オクターブ8重和音FM音源搭載●オートロード・オートイジェクトの1Mバイト5インチFDD2基搭載●マウス・トラックボール標準装備

●写真はCZ-644C-TNとCZ-614D-TN。

本体＋キーボード＋マウス・トラックボール

CZ-634C-TN（チタンブラック）　標準価格368,000円（税別）

81MB HDタイプ　CZ-644C-TN（チタンブラック）　標準価格518,000円（税別）

SUPER 本体＋キーボード＋マウス・トラックボール

CZ-604C-TN（チタンブラック）　標準価格348,000円（税別）

81MB HDタイプCZ-623C-TN（チタンブラック）　標準価格498,000円（税別）

PROII 本体＋キーボード＋マウス

CZ-653C-BK（ブラック）・-GY（グレー）　標準価格285,000円（税別）

40MB HDタイプCZ-663C-BK（ブラック）・-GY（グレー）　標準価格395,000円（税別）

- 15型カラーディスプレイテレビ（ドットピッチ0.39mm）　CZ-605D-BK（ブラック）・-GY（グレー）　標準価格115,000円（スピーカー2個/チルトスタンド同梱・税別）※
- 14型カラーディスプレイテレビ（ドットピッチ0.31mm）　CZ-607D-TN（チタンブラック）・-BK（ブラック）　標準価格99,800円（チルトスタンド同梱・税別）***NEW***
- 15型カラーディスプレイテレビ（ドットピッチ0.31mm）　CZ-614D-TN（チタンブラック）・-BK（ブラック）　標準価格135,000円（スピーカー2個/チルトスタンド同梱・税別）***NEW***
- 14型カラーディスプレイ（ドットピッチ0.31mm）　CZ-604D-BK（ブラック）・-GY（グレー）　標準価格94,800円（スピーカー2個/チルトスタンド同梱・税別）
- 14型カラーディスプレイ（ドットピッチ0.31mm）　CZ-606D-TN（チタンブラック）・-BK（ブラック）・-GY（グレー）　標準価格79,800円（チルトスタンド同梱・税別）
- 21型カラーディスプレイ（ドットピッチ0.52mm）　CU-21HD（ブラック）　標準価格148,000円（スピーカー2個同梱・税別）

※印の商品は在庫僅少です。

地区予選大会開催中!! お友達を連れてぜひ、ご来場ください。

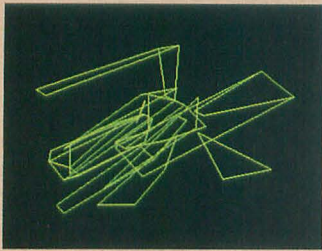
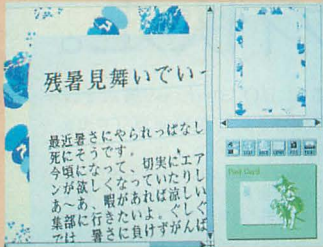
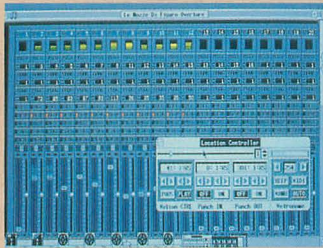
開催日	開催地	会場	応募・問い合わせ先
9月8日(日)	東北地区	シャープ仙台ビル 4Fホール 仙台市若林区卸町東3-1-27 ☎022-288-9111	シャープエレクトロニクス販売㈱ 東北統轄(営)パソコン担当 ☎022-288-9111㈹
9月15日(日)	中国地区	広島市西区民センター 3F大会議室A 広島市西区横川新町6-1 ☎082-234-1960	シャープエレクトロニクス販売㈱ 中国統轄(営)パソコン担当 ☎082-874-2282㈹
9月22日(日)	北関東地区	護国会館 平安の間 宇都宮市陽西町1-37 ☎0286-22-3180	シャープエレクトロニクス販売㈱ 北関東統轄(営)パソコン担当 ☎0286-35-1151㈹

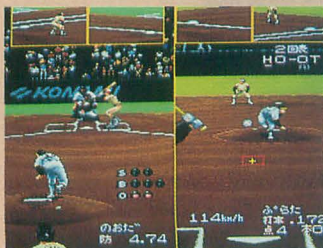
●お問い合わせは…

シャープ株式会社

電子機器事業本部システム機器営業部
〒545 大阪市阿倍野区長池町22番22号☎(06)621-1221（大代表）
電子機器事業本部液晶映像システム事業部第2商品企画部
〒162 東京都新宿区市谷八幡町8番地☎(03)3260-1161（大代表）

C O N T

〈スタッフ〉
●編集長／前田 徹 ●副編集長／植木章夫 ●編集／岡崎栄子 浅井研二 山田純二 ●協力／有田隆也 中森 章 林 一樹 吉田幸一 華門真人 毛内俊行 吉田賢司 影山裕昭 古村 聡 村田敏幸 丹 明彦 三沢和彦 長沢淳博 宮島 靖 金子俊一 浦川博之 石上達也 ●カメラ／杉山和美 ●イラスト／永沢しげる 山田晴久 小栗由香 ●アートディレクター／島村勝頼 ●レイアウト／元木昌子 ADGREEN ●校正／グループごじら

特集 Brush up your MAGIC.

NEW Print Shop PRO-68K Ver2.0

Musicstudio Mu-I Super

イース

生中継68

DōGA CGアニメーション講座

表紙絵：塚田　哲也

E N T S

1991 SEP.
9

■広告目次

SHARP

シャープ株式会社 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)

入力から、編集、出力、複写まで、カラーの世界は、シャープです。

SHARP IS COLOR

▶INPUT

X68000用パラレルインターフェイスを標準装備した
高速コンパクト型イメージスキャナ。

カラーイメージスキャナ JX-220X……標準価格168,000円(税別)

●A4サイズの原稿を約50秒※1で高速読み取り●CCDセンサー採用。さらに中間調処理でシャープでリアルな画像を再現●ディザパターン指定機能※2や濃度補正機能※2など高度な画像処理機能で緻密な読み取りが可能●解像度200ドット/インチ(約7.9ドット/mm)。ズーム機能で1%きざみの拡大、縮小も可能●色ずれの少ない線順次(1走査)読み取り●X68000シリーズ用「スキャナツール」ソフトを標準装備●プリンタと直接接続することによりダイレクトプリント※3が可能●RS-232Cインターフェイス/X68000シリーズ用専用パラレルインターフェイスを標準装備。

※1:A4、2値出力、コンピュータへの実転送時間。
※2:表記機能はJX-220X本体使用であり、付属ユーティリティ使用時は異なります。
※3:別売のパラレルインターフェイスケーブル(JX-22PC標準価格12,000円(税別))が必要です。

▶OUTPUT

3種類の制御コマンドモードを搭載。
質感も鮮やかに再現する高品位カラーイメージジェット。

カラーイメージジェット IO-735X-B……標準価格248,000円(税別)

●シャープ独自のIOシリーズコマンド(Gモード)に加え、NM-9900モード(Nモード)、ESC/P24-84C準拠モード(Pモード)をサポート。一般文書の作成から、各種デザイン、建築用パースなどのCAD分野に対応●発色性に優れた普通紙対応の新黒インキ採用。専用紙はもちろんオフィスでよく使われる普通紙にも鮮明カラー印字●プリントバッファメモリ(128KB)の内蔵で、ホストコンピュータの拘束時間を軽減●48ノズル(各色12ノズル)採用の高速印字。A4-1ページを※約90秒でプリント(データ受信時間除く)●ビジネス用途に適したB4横用紙幅対応●OHPフィルム(専用)にも鮮明プリント●ノンインパクト方式ならではの静粛印字●インキ補充は簡単、経済的なカートリッジ方式

※261×174mm領域

IO-735X-B 対応アプリケーション

●SX-WINDOW対応ペイントツール
Easypaint SX-68K
CZ-263GW 標準価格12,800円(税別)

●WYSIWYGを実現、ドローグラフィックソフト
CANVAS PRO-68K
CZ-249GS 標準価格29,800円(税別)

●オリジナリティを活かせるポップアップツール
NEW Printshop PRO-68K ver2.0
CZ-221HS 標準価格20,000円(税別)

●マルチワープロ PRO-68K
Multiword
CZ-225BS 標準価格32,000円(税別)

●高速カード型リレーショナルデータベース
CARD PRO-68K ver2.0
CZ-253BS 標準価格29,800円(税別)

●パソコン通信もできるメモリ常駐型ソフト
Teleportion PRO-68K
CZ-258BS 標準価格22,800円(税別)

●これからの高速通信をサポート
Communication PRO-68K ver2.0
CZ-257CS 標準価格19,800円(税別)

平成3年9月末日迄
(バナナキャンペーン実施中!)
期間中IO-735X-Bを買うと便利なプリントツール(デモ版)がついてくる。

■拡大縮小、マルチ印刷など多彩な印刷機能を装備したプリントツール
BANANA PRINT………………標準価格48,000円(税別)〈発売元:㈲ムーンベース〉☎022(271)9700

資料のご請求・お問い合わせはコンシューマーセンター/東日本OA相談室 〒162 東京都新宿区市谷八幡町8番地 ☎(03)3260-1161(大代表)・西日本OA相談室 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)・名古屋OA相談室 〒454 名古屋市中川区山王三丁目5番5号 ☎(052)332-2611(大代表)・福岡OA相談室 〒816 福岡市博多区井相田2丁目12番1号 ☎(092)575-2381(代表)

MONTHLY PICK UP

●シューティングゲーム

中華大仙

CZ-268AS　標準価格7,900円(税別)

©TAITO CORP. 1988

●コミカルアクションゲーム

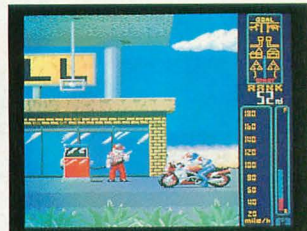
ボナンザブラザース

CZ-270AS　開発中

©SEGA1990 REPROGRAMMED BY SHARP
●この写真は、イメージ写真です。

●バイクレーシングゲーム

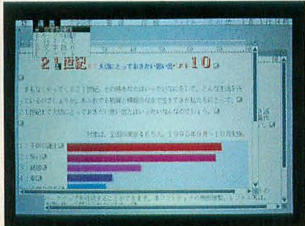
ダッシュ野郎

CZ-269AS　標準価格8,800円(税別)

©TOAPLAN Co. Ltd. 1988

●ウィンドウでWYSIWYG編集

マルチワープロ PRO-68K Multiword

8月発売予定

CZ-225BS 標準価格32,000円(税別)

マルチウィンドウを駆使したウィンドウモードと、素早い編集が可能な高速テキストモードを装備。カラーグラフィックも自由にレイアウト。レーザープリンタにも対応したマルチワープロです。

※メインメモリ2MB必要です。

●Zeit日本語ベクトルフォントをサポート

8月発売予定

CZ-265HS 標準価格20,000円(税別)

処理速度の高速化はもちろん、カセットレーベル、カレンダー作成に対応したほか、モノクロデータの編集などグラフィックエディタを強化した高機能テキストエディタを内蔵しています。

※メインメモリ2MB必要です。
※NEW Print Shop PRO-68K(CZ-221HS)をお持ちの方には有償バージョンアップを行います。

●メモリ常駐型の優れモノ

Teleportion PRO-68K

CZ-258BS 標準価格22,800円(税別)

他のソフトを実行中でも任意に呼び出して使えるメモリ常駐型のソフト。パソコン通信/エディタ/スケジュール/住所録/メモ帳などの機能を文具感覚で使えます。

※メインメモリ2MB必要です。
※Stationery PRO-68K(CZ-240BS)をお持ちの方には有償バージョンアップを行います。

●高速カード型リレーショナルデータベース

CZ-253BS 標準価格29,800円(税別)

操作性の向上、高速化を図った新マルチウィンドウシステムを搭載したニューバージョンです。一覧表画面入力、グラフ機能などをサポート。キーボード操作にも対応します。

※メインメモリ2MB必要です。
※CARD PRO-68K(CZ-226BS)をお持ちの方には有償バージョンアップを行います。

《お詫びと訂正》 ■弊社発行のX68000ソフト情報誌「ソフトウェアフィールド」20号において、一部標準価格に誤りがありますので訂正させていただくとともに、謹んでお詫び申し上げます。

- ●Musicstudio PRO-68K ver2.0(CZ-261MS)……………(誤)標準価格 18,800円(税別)→(正)標準価格 28,800円(税別)
- ●中華大仙(CZ-268AS)……………(誤)標準価格 8,800円(税別)→(正)標準価格 7,900円(税別)
- ●光磁気ディスクユニット(CZ-OMO1)……………(誤)標準価格 29,800円(税別)→(正)標準価格450,000円(税別)
- ●SCSIボード(CZ-6BS1)……………(誤)標準価格450,000円(税別)→(正)標準価格 29,800円(税別)

※CZ-258BS、CZ-265HS、CZ-253BS、CZ-278SSの有償バージョンアップについては、下記にお問い合わせください。

●お問い合わせは…シャープ㈱電子機器事業本部液晶映像システム事業部第2商品企画部 〒162 東京都新宿区市谷八幡町8番地 ☎(03)3260-1161(大代表)へ。シャープ株式会社

SHARP システムパフォーマンスを実証する多彩なペリフェラル。

ディスプレイ関連

カラーディスプレイテレビ

14型カラーディスプレイテレビ
CZ-607D-BK・-TN
標準価格99,800円(税別)
(チルトスタンド同梱)

15型カラーディスプレイテレビ
CZ-605D-BK・-GY
標準価格115,000円(税別)
(スピーカー2個・チルトスタンド同梱)

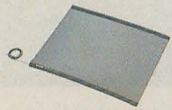
15型カラーディスプレイテレビ
CZ-614D-BK・-TN
標準価格135,000円(税別)
(スピーカー2個・チルトスタンド同梱)

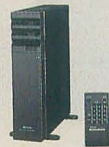
カラーディスプレイ

14型カラーディスプレイ
CZ-606D-TN・-BK・-GY
標準価格79,800円(税別)
(チルトスタンド同梱)

14型カラーディスプレイ
CZ-604D-BK・-GY
標準価格94,800円(税別)
(スピーカー2個・チルトスタンド同梱)

21型カラーディスプレイ
CU-21HD
標準価格148,000円(税別)
(スピーカー2個同梱)

CRTフィルター

高性能CRTフィルター
BF-68PRO
標準価格19,800円(税別)
(14/15型用)

チューナー

RGBシステムチューナー
CZ-6TU-BK・-GY
標準価格33,100円(税別)
(リモコン付)

アートツール

画像入力

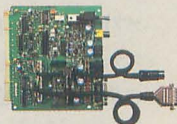
カラーイメージスキャナ※1
CZ-8NS1
標準価格188,000円(税別)

カラーイメージスキャナ※1
JX-220X
標準価格168,000円(税別)
※RS-232C/パラレルインターフェイス標準装備

スキャナ用パラレルボード
CZ-6BN1
標準価格29,800円(税別)

映像入力

カラーイメージユニット※2
CZ-6VT1-BK
CZ-6VT1
標準価格69,800円(税別)

映像出力

ビデオボード※3
CZ-6BV1
標準価格21,000円(税別)

プリンタ

熱転写カラープリンタ

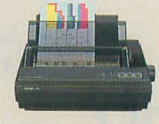
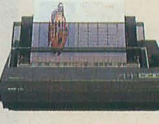
48ドット
熱転写カラー漢字プリンタ
CZ-8PC5-BK
標準価格96,800円(税別)

カラービデオプリンタ

カラービデオプリンタ
★CZ-6PV1
標準価格198,000円(税別)
(信号ケーブル同梱)

カラーイメージジェット

カラーイメージジェット※4
IO-735X-B
標準価格248,000円(税別)
(信号ケーブル別売)
※グレータイプのIO-735Xもあります。

カラードットプリンタ

24ピン
カラー漢字プリンタ(80桁)
CZ-8PG1
標準価格130,000円(税別)
(信号ケーブル同梱)

24ピン
カラー漢字プリンタ(136桁)
CZ-8PG2
標準価格160,000円(税別)
(信号ケーブル同梱)

ドットプリンタ

24ピン漢字プリンタ(136桁)
CZ-8PK10
標準価格97,800円(税別)
(信号ケーブル同梱)

ファイル

光磁気ディスク

光磁気ディスクユニット※5
(594MB)
CZ-6MO1
標準価格450,000円(税別)
(SCSIケーブル同梱)
※光磁気ディスクカートリッジは別売です。別売のJY-701MPA 標準価格30,000円(税別)をご使用ください。

ハードディスク

増設用ハードディスクドライブ (40MB)
(CZ-602C/603C/652C/653C内蔵用)
★CZ-64H※
標準価格120,000円(税別)
(取付費別)

増設用ハードディスクドライブ (81MB)
(CZ-604C/634C内蔵用)
CZ-68H※
標準価格160,000円(税別)
(取付費別)

※取付に関してはシャープお客様ご相談窓口にてご相談ください。

ハードディスクユニット(20MB)
★CZ-620H
標準価格178,000円(税別)
※CZ-604C/623C/634C/644Cでは使用できません。

※1 ご使用に際しては、カラーイメージスキャナCZ-8NS1、JX-220Xに同梱のRS-232Cケーブルで接続するか、より高速のパラレルデータ伝送を行う場合、別売のスキャナ用パラレルボードCZ-6BN1標準価格29,800円(税別)で接続してください。※2 テレビチューナーを内蔵していないディスプレイをご使用の場合は、RGBシステムチューナーCZ-6TU(別売)が必要です。※3 ビデオ出力は15.75kHzテレビ標準信号です。また、拡張I/Oスロットは2スロット使用します。※4 別売の信号ケーブルIO-73CX標準価格5,500円(税別)で接続してください。※5 CZ-600C、601C、602C、603C、611C、612C、613C、652C、653C、662C、663Cにご使用の場合は、別売のSCSIボード(CZ-6BS1)が必要です。また、X68000用 OS Human 68k ver 2.0以上にてご使用ください。(光磁気ディスクカートリッジは別売のJY-701MPA 標準価格30,000円(税別)をご使用ください。) ※6 ご使用に際しては、あらかじめ別売の1MB増設RAMボードCZ-6BE1 標準価格 35,000円(税別・

シャープ株式会社 ●お問い合わせは…シャープ㈱電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号☎(06)621-1221(大代表

資料請求券 X68000ペリフェラル Oh!X 9月

宇宙は、野望だけでは支配できない。

宇宙暦796年、銀河系はゴールデンバウム王朝が支配する銀河帝国と、その専制政治に反対する自由惑星同盟の両陣営が激しい戦闘を繰り返していた……。圧倒的支持を得た「銀河英雄伝説」を遥かに凌ぐスケールで、今新たな伝説が生まれようとしている。銀河英雄伝説IIだ。帝国軍の若き天才ラインハルト、そしてヤン・ウェンリーの熱い闘いが、再び始まる。星系マップは従来の4倍、3Dグラフィックによる戦闘シーンなど、あらゆる面でパワーアップされている。田中芳樹原作の大人気スペースオペラ「銀河英雄伝説」。宇宙の歴史を変える闘いは、ここに始ろうとしている。

SPACE WAR SIMULATION

銀河英雄伝説II

DX⁺kit

デラックスプラスキット

X68000シリーズ(3枚組) 8月30日新発売! ¥5,000(税別)

●プレイするには「銀河英雄伝説II」が必要です。

- ●キャンペーンモード付新シナリオ10本
- ●最大ユニット数・艦隊数の増強
- ●提督・部隊の初期配備エディター
- ●艦隊の部隊構成エディター
- ●思考ルーチンの強化および高速化
- ●新提督の登場
- ●難易度の設定ができる
- ●拡張ルールでプレイ
- ●旗艦アニメーションの追加
- ●BGMにボレロ、悲愴、英雄を追加

開発中の画面

銀河英雄伝説IIDX⁺set X68000シリーズ(7枚組)9月6日発売 ¥12,800(税別)

※通信販売(送料サービス):品名・機種・住所・氏名・電話番号を明記して、現金書留でお申し込みください。

BOTHTEC® ボーステック・ソフトウェア
株式会社クエスト 〒158 東京都世田谷区用賀2-18-8 TEL.03(3708)4711

あなたの夢物語、見せてください。

第2回ブルトン・レイ シナリオコンテスト作品募集

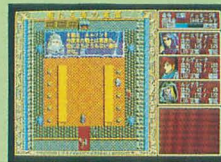
コンプティーク・システムソフト共同主催による、
「第2回ブルトン・レイシナリオコンテスト」を開催します。
対象機種はX68000シリーズ、及びPC-9800シリーズです。
詳しい内容につきましては、月刊コンプティーク誌7・8・9月号をご覧ください。
皆様から、多数の作品応募をお待ちしています。

ファンタジーの枠を超えて、X68000ワールドに広がるブルトン・レイ ファミリー。

ブルトン・レイ

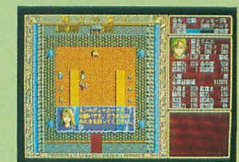
剣と、魔法と、ロマンの小宇宙。

中世ヨーロッパを舞台に、英雄達の冒険を描く本格派ロールプレイングゲーム。小気味よいテンポで進む短編シナリオの数々が、剣と魔法とロマンに満ちたファンタジーの小宇宙へと案内する。

〈好評発売中〉
■X68000シリーズ
■5"-2HD(3枚組)
●アナログRGB(31KHz対応)ディスプレイをお使いください。
価格8,800円

ブルトン・レイ シナリオ集

珠玉のシナリオアンソロジー。

多彩な十篇の物語を集めた、シナリオ・アンソロジー。短編シネマのようなシナリオの数々が、ブルトン・レイの世界をさらに広げる。

〈好評発売中〉
■PC-9800シリーズ
■5"-2HD(2枚組)
●別売の「ブルトン・レイ」が必要です。
※PC-9800シリーズ用となっておりますが、X68000シリーズでもご使用いただけます。
※3.5"-2HDのディスクは、PC-9800シリーズ用です。
価格4,800円

ブルトン・レイ シナリオエディタ

思い描いていた物語が、動き出す。

ブルトン・レイのシナリオを、思いのままに自由に創ることができるシナリオエディタ。ストーリーメイキングのための多彩な機能で、誰にでも自分だけの世界が創造できるクリエイティブ・ツールだ。

〈好評発売中〉
■X68000シリーズ
■5"-2HD(2枚組)
●別売の「ブルトン・レイ」X68000版が必要です。
●アナログRGB(31KHz対応)ディスプレイをお使いください。
●メインメモリが2MB以上の場合、日本語入力フロントプロセッサとしてASK68Kが使用できます。
(2MB未満の場合は、単漢字入力となります。)
価格5,800円

コンプティーク ブルトン・レイ シナリオコンテスト入賞作品

ブルトン・レイ シナリオ集vol.2

誰も知らない、世にも新奇な物語。

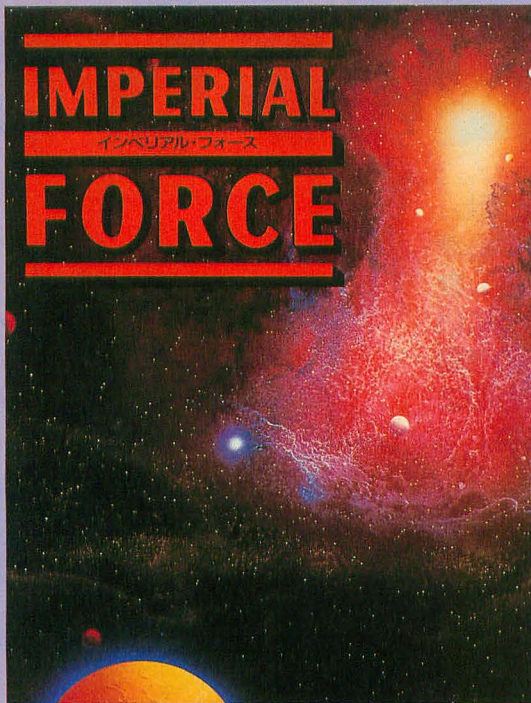
コンプティーク誌上で行われた"シナリオコンテスト"の応募作から、特に優れた作品、ユニークな物語を選りすぐったシナリオ集第2弾。

〈好評発売中〉
■PC-9800シリーズ
■5"-2HD(2枚組)
●別売の「ブルトン・レイ」が必要です。
※PC-9800シリーズ用となっておりますが、X68000シリーズでもご使用いただけます。
※3.5"-2HDのディスクは、PC-9800シリーズ用です。
価格4,800円

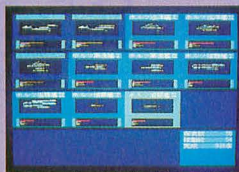
●システムソフトアミューズメント事業部では、パソコンネットをテスト開局しております。現在、会員募集はしておりませんので、アクセスはゲストIDにてお願いします。テレフォンサービスともどもご利用ください。

電話番号……03-3326-9644
通信速度……1200～2400MNP5
ビット長………8
パリティ…………NO
ストップビット……1
漢字……………SHIFT-JIS漢字

新製品の発売日および内容のご案内は…
テレフォンサービス専用電話　東京:03-3326-8710
福岡:092-752-2602
アミューズメント事業部営業部専用電話
092-752-5262(祝祭日を除く月～金)

◉総合カタログをご希望の方は請求券をはがきに貼り、住所・氏名・年齢・電話番号・使用機種名を明記の上、弊社宛にご送付ください。

インペリアル・フォース

戦略は宇宙空間へ。

戦略ゲームの面白さを宇宙空間にまで広げた、SFシミュレーション「インペリアル・フォース」がX68000に登場。舞台となる銀河宇宙を自動作成するマップジェネレータやリアルタイムオペレーション、フォーメーションバトルなどの先進システムはそのままに、グラフィックを全面的に強化。よりリアルなバトルシーンが、X68000上に展開される。

※画面は開発中のものです。

〈9月発売予定〉
■X68000シリーズ
■5"-2HD(2枚組)
●アナログRGB(31KHz対応)ディスプレイをお使いください。
●入力装置として、X68000シリーズ添付のマウスを使用します。

価格8,800円

リアルタイム・シミュレーションの波が、X68000を直撃する!

大戦略III'90

臨場感、深まる。

シリーズ最高峰に君臨する「大戦略III'90」が、満を持してX68000に堂々のデビューを果たす。ビジュアル、サウンド、操作性のすべてにわたってグレードアップしたのに加え、PC-98シリーズのマップ、ゲームデータの活用も可能になった。ゲームをさらに盛り上げるBGMや効果音も加わり、臨場感はさらに深まる。

※画面は開発中のものです。

〈10月発売予定〉
■X68000シリーズ ■5"-2HD(3枚組)
●アナログRGB(31KHz対応)ディスプレイをお使いください。
●入力装置として、X68000シリーズ添付のマウスを使用します。
※「大戦略III'90」X68000版に限りまして、技術的な内容などユーザーサポートに関するお問い合わせはアルシスソフトウェア、販売に関するお問い合わせはシステムソフト営業部までお願いいたします。

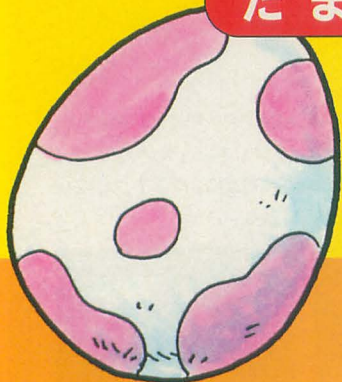
㈲アルシスソフトウェア
佐世保市松浦町5-13 グリーンビル3F
〒857 TEL0956-22-3881

価格9,800円

■広告の商品は、全国のパソコンショップでお求めになれます。
■通信販売をご希望の場合は、現金書留、郵便為替、または郵便振替(福岡3-37311)で、商品名、機種、メディア名、および連絡先の電話番号を明記の上、送料を添えて弊社までお申し込みください。表示価格に消費税は含まれておりませんので、価格の3%の消費税も併せてお願いいたします。送料は消費税込みで一律400円です。(送料のみ切手でも可)
■お申し込みから商品がお手許に届くまで、通常、2~3週間ほどかかります。
■商品に欠陥がある場合を除いて、お届けした商品の返品はご容赦願います。

※製品の仕様は、機能・性能の改善のため将来予告なしに変更することがあります。
※表示価格に消費税は含まれておりません。

SystemSoft 株式会社 システムソフト
アミューズメント事業部
〒810 福岡市中央区天神3丁目10-30

総合カタログVol.11 請求券
Oh!X 9月号
有効期限:1991年9月末

たまご

シードラゴン

チャイルドドラゴン

聖竜伝説

モンビット

MONBIT

8/30発売　6,800円

HUDSON GROUP
HUDSON SOFT®

本　　社　〒062 札幌市豊平区平岸3条5丁目1番18号 ハドソンビル TEL 011-841-4622
東京支社　〒162 東京都新宿区市谷田町3丁目1番1号 ハドソンビル TEL 03-3260-4622
大阪支店　〒542 大阪市中央区東心斎橋1丁目1番10号 大阪料理会館ビル5階 TEL 06-251-4622
営 業 所　札幌・名古屋・福岡

PCエンジン CD-ROMシステム
ROM ROM
ぽけな僕たち
進化するドラゴン、
最後はどう変わるのか!?
スカイドラゴン
ファイヤードラゴン
モンビットのエンディングテーマ曲
「青春はちっぽけな僕たち」もよろしくね
「モンビット」のエンディングテーマがCDシングルに/ 歌うのはこの曲でデビューの井上麻美ちゃん。ミスCDロムのグランプリに輝いた美少女だぞ。8月25日、ファンハウスから発売!!
さらに拡がるPCエンジンの世界
TENNOKOE BANK
天の声バンク
データバックアップカード
"天の声2"や"CD・ROM² インターフェイス(IFU30)"等のバックアップ・データを、最大4台分も保存できる。カードサイズで持ち運びにも便利で、ゲームのデータを気軽に出し入れすることが可能だ。
定価3,880円 9月6日発売
TENNOKOE BANK
天の声バンク
© 1991 HUDSON SOFT
HuCARD

全国通販

SHARP 認定 PPO-SHOP **O.A.ランド**

(TEL) 03-3770-8855

■アフターサービス万全のサポート体制
●下取・買取は電話で見積りしております。責任を持って下取りさせて頂きます。

営業時間
平日………AM10:00〜PM7:00
土日・祭日…AM10:00〜PM6:00

▶8・15〜9・18

流通事情により、広告表示価格は、お安くなる場合がありますので、ドンドンお電話下さい。

CYBER STICK
■CZ-8NJ2
(定価¥23,800)
OAランド特価 ▶¥18,000

電子手帳 ●見やすい漢字4桁表示!! 情報化時代の必需品!!
■PA-9500(¥48,000)…▶特価¥38,000
■PA-8500(¥28,000)…▶特価¥15,000
■PA-7500(¥22,000)…▶特価¥12,000

SHARPのことなら なんででおまかせ!!

大徳買セール! 安く値切ってネ。(本体セット:送料 消費税込み)
お電話下さい。㊙価格をお知らせいたします。

SHARP X68000シリーズセット(送料・消費税込み)

X68000XVI

①CZ-634-TN+CZ-614D-TN
定価合計¥503,000

回数	月額
12回	¥33,100
24回	¥17,600
36回	¥12,200
48回	¥ 9,600

②CZ-634C-TN+CZ-607D-TN
定価合計¥467,800

回数	月額
12回	¥30,800
24回	¥16,300
36回	¥11,400
48回	¥ 8,900

③CZ-634C-TN+CZ-606D-TN
定価合計¥447,800

回数	月額
12回	¥29,500
24回	¥15,700
36回	¥10,900
48回	¥ 8,500

X68000XVI-HD

①CZ-644C-TN+CZ-614D-TN
定価合計¥653,000

回数	月額
12回	¥42,800
24回	¥22,700
36回	¥15,800
48回	¥12,400

②CZ-644C-TN+CZ-607D-TN
定価合計¥618,700

回数	月額
12回	¥40,600
24回	¥21,500
36回	¥15,000
48回	¥11,700

③CZ-644C-TN+CZ-606D-TN
定価合計¥597,800

回数	月額
12回	¥39,300
24回	¥20,800
36回	¥14,500
48回	¥11,400

■CZ-634C■ 特価 ¥TEL下さい!!

■CZ-644C■ 特価 ¥TEL下さい!!

XVIお買い上げの方に❶ニュージーランドストーリー ❷V-BALL ❸ジョイカード(連射式) ❹ディスケット20枚プレゼントいたします!!

現金でお買い上げの方には、さらに超特値でお出ししてます。ぜひ一度TEL下さい!!

上記組合せのディスプレイ(モニター)変更自由!!
詳しくは、お電話にてお問い合せ下さい!!

★FUJITSU FM TOWNSシリーズ

①FM-TOWNS 20F
FM-TOWNS 20F
FMT-DP533
FMT-KB105
定価¥422,800
特価¥268,000

②FM-TOWNS 40H
FM-TOWNS 40H
FMT-DP533
FMT-KB105
定価¥573,100
特価¥378,000

③FM-TOWNS 80H
FM-TOWNS 80H
FMT-DP533
FMT-KB105
定価¥722,800
特価¥475,000

本体、単体でも販売してます。くわしくは、TEL下さい!!

周辺機器コーナー 電話で値切ろう。

プリンターセットコーナー

①CZ-8PC5 NEW 定価¥96,800
●48ドット●熱転写カラー 漢字プリンター
大特価TEL下さい!!

②CZ-8PK10(24ピン漢字プリンター136桁)
定価¥97,800……特価¥71,000

③CZ-8PG1(24ピンカラー漢字プリンター80桁)
定価¥130,000…特価¥93,000

④CZ-8PG2(24ピンカラー漢字プリンター136桁)
定価¥160,000…特価¥114,000

OAランド特選品!!

■IO-735XB(定価¥248,000)
●カラーイメージジェットプリンター ケーブル付 特価¥169,000

X68000用ハードディスク

■SCSIタイプ TOWNSでもOK
●アイテック
①TX-80S(¥108,000)…特価¥ 78,500
②TX-130S(¥138,000)…特価¥ 98,500
③TX-180S(¥185,000)…特価¥132,000

■SASIタイプ
●アイテック
①ITX-680(¥198,000)……特価¥ 68,000
●ロジテック
①SHD-40(¥99,800)……特価¥ 60,000

※X68000SUPER/XVI以外の機種では、SCSIボードが必要となります。
★SCSIボード…………特価¥ 22,000
★光ディスク…………特価¥320,000

X68000用周辺機器コーナー

①CZ-6VT1(カラーイメージユニット)
定価¥69,800……特価¥ 51,500
②CZ-8NS1(カラーイメージスキャナー)
定価¥188,000…特価¥135,000
③CZ-6BM1(MIDIボード)
定価¥26,800……特価¥ 20,000
④CZ-6BE2A(2MB増設RAMボード)
定価¥59,800……特価¥ 44,000
⑤CZ-6BE2B(2MB増設RAM)
定価¥54,800……特価¥ 40,500
⑥CZ-6BP2(数値演算プロセッサ)
定価¥45,800……特価¥ 33,800
⑦CZ-6EB1(拡張I/Oボックス=4スロット)
定価¥88,000……特価¥ 65,000
⑧CZ-6BP1(数値演算プロセッサボード)
定価¥79,800……特価¥ 59,000

《計測技研》増設メモリ&プロセッサ

●高速増設メモリと数値演算プロセッサが一つのボードになった!!●

●KGB-X68PRKII-02(¥ 55,000)……特価¥ 42,800
●PRKII-04(¥ 90,000)……特価¥ 70,200
●PRKII-06(¥125,000)……特価¥ 97,500
●PRKII-08(¥160,000)……特価¥124,800
●PRKII-12(¥ 85,000)……特価¥ 66,300
●KGB-X68PRKII-14(¥120,000)……特価¥ 93,600
●PRKII-16(¥155,000)……特価¥121,000
●PRKII-18(¥190,000)……特価¥148,000
●MC-6888 IRC(¥38,000)…………特価¥ 28,500

I・Oデータ増設RAMボード

■PIO-6BE1-A (1MB) 定価¥25,000 特価¥17,300
■PIO-6BE2-2M (2MB) 定価¥50,000 特価¥33,500
■PIO-6BE4-4M (4MB) 定価¥88,000 特価¥58,500

★OAランド推奨ソフト

Ⓐ Easy Paint SX 68K (CZ-263GW) 特価TEL下さい!!
Ⓑ Music studio PRO 68K (CZ-261MS) 定価¥28,800 特価TEL下さい!!
Ⓒ Print Shop V.2 (CZ-265HS) 特価TEL下さい!!
Ⓓ Multiword PRO 68K (CZ-225BS) 定価¥32,000 特価¥24,000
Ⓔ CZ-245LS (C-コンパイラII) 定価¥44,800 特価¥33,500
Ⓕ Telepotion PRO 68K (CZ-258BS) 定価¥22,800 特価¥18,000

通信販売のご案内

—— 全国通販 ——

■銀行振込で申し込みの方は商品名及びお客様の住所・氏名・電話番号をお知らせ下さい。
〔振込先〕第一勧業銀行 渋谷支店
普通No.1163457 ㈱オーエーランド

■年中無休です!!

■現金書留で送金されるお客様は電話番号と商品名、数量を明記して同封して下さい。■クレジットでご購入を希望される方は申し込み用紙をお送り致しますのでご記入の上返送して下さい。20才以上の方は、原則として保証人不要です。クレジットは1〜60回払で月々5,000円よりご自由に設定できます。

東専京門工学業校
東急観光
至三軒茶屋246号
大和證券
ヤマハエピキュラス
東急プラザ
京王井ノ頭線
歩道橋
南口バスターミナル
JR渋谷南口
至恵比寿
至六本木
ココです!! NEW OAランド
(・ニコマートの2Fです。)

クレジット表

回数	率	回数	率	回数	率	回数	率	回数	率	回数	率	回数	率
3回	3.5%	6回	4.5%	10回	6%	12回	6%	15回	8.5%	18回	11%	20回	12%
24回	12.5%	30回	17%	36回	17.5%	42回	22.5%	48回	23%	54回	29%	60回	29.5%

㈱オー・エー・ランド

〒150 東京都渋谷区桜丘町3-13 アルカディア2F
☎(03)3770-8855
関東エリアの送料は、1個につき¥1,000です。 FAX(03)3770-7080
★全商品保証書付。専門のアドバイザーが、お客様のニーズに対応します。
★初期不良・輸送トラブル等に迅速に対応し、即交換させていただきます。

■表示価格は、税別表示です。詳しくは、お電話にて、お問い合せ下さい。掲載の価格は、6月 下旬現在です。

低金利クレジットをご利用下さい。平日AM10時〜PM7時、土日・祭日AM10時〜PM6時迄ガンバッテま〜す!!

■ゲームソフト、ビジネスソフト2割・3割引きは、当り前で〜す。お問い合せ下さい。

待たせたな!!

大変長らくお待たせいたしました。X68000ユーザーお待ちかねの「ロードス島戦記」が、ついに発売されます。
アニメビデオや小説版でも大人気の「ロードス島戦記」の世界を、パーンやディードリットと共に思う存分お楽しみ下さい。

X68000
8月24日
新発売

ロードス島戦記©
灰色の魔女

原作：安田　均・水野　良
オリジナルキャラクターデザイン：出渕　裕　標準価格 **9,800円**

【ユーザーズテレホン　☎大阪06(315)8255
平日の午後1時半から6時の間は、お問い合せに直接お答えします。
その他の時間と土・日・祝日は、まるまる24時間録音できるテープサービスです。】

■標準価格に消費税は含まれておりません。お買い上げの際に別途消費税をお支払い下さい。■通信販売ご希望の方は、住所・氏名・電話番号・商品名・機種名・メディアを明記の上、現金書留または郵便振替（大阪8-303340）にてお申し込み下さい。送料は無料ですが、標準価格に消費税の3%を加えた金額をお送り下さい。

株式会社エム・エー・シー ハミングバードソフト
〒530 大阪市北区曽根崎2丁目2番15号

超低金利クレジットOK

注目!!

冬のボーナス一括払い
手数料(金利)無料
(平成3年12月末をご利用下さい。)

またまた 秋葉原で おなじみの

8/15〜9/15

●お近くの方はお
●本体単品で特
●ビジネスソフト定

X6800用ハードディスク(80M)
■TX-80(アイテック) SASI SCSI 両用
定価¥108,000 特価¥78,000
(送料・消費税込み¥81,370)

Fine Scanner-X68
(HAL研究所)X68000専用
■HGS-68(定価¥39,800)
特価¥25,500
(送料・消費税込み¥27,295)

X68000シリーズ専用
MIDIインターフェースボード
SX-68M(サコム)
(純生コンパチ)定価¥19,800
特価¥13,900
(送料・消費税込み¥14,832)

X68000メモリボード(シャープ&I/O・DATA)(送料¥500)
①CZ-6 BE1(600C用)定価¥35,000 (送料・消費税込¥27,295)… 特価¥26,000
②PIO-6BE1-A 定価¥25,000 (送料・消費税込み¥17,201)・特価¥16,200
③PIO-6BE2-2M 定価¥50,000 (送料・消費税込み¥33,475)・特価¥32,000
④PIO-6BE4-4M 定価¥88,000 (送料・消費税込み¥57,680)・特価¥55,500

ジョイスティック 送料¥500
●X-1PRO
定価¥9,500▶特価¥7,800
●ASCII STICK
定価¥6,800▶特価¥5,500

P&A超低金利クレジットをご利用ください!!

X68000-XVI

※クレジット表は、送料・消費税込み!!

XVI/XVI-HDセットでお買い上げの方に、もれなくプレゼント!!
①V'BALL(¥7,900)
②熱血高校サッカー編(¥8,800)
③ダウンタウン熱血物語(¥8,800)
の他に、さらにその上人気の
Ⓐ「パロディハウスだ(¥9,800)」又は
Ⓑ「ファランクス(¥8,800)」のどちらか1本をプレゼント!!

X68000-XVI ▶セットでお買い上げの方に●ディスケット10枚●ジョイカード2ケプレゼント中!!

Ⓐセット:CZ-634C-TN+CZ-606D-TN…定価¥447,800▶特価価格はTEL下さい。

12回	29,200	24回	15,500	36回	10,800	48回	8,500	60回	7,100

Ⓑセット:CZ-634C-TN+CZ-614D-TN …定価¥503,000▶特価価格はTEL下さい。

12回	32,800	24回	17,400	36回	12,100	48回	9,500	60回	8,000

X68000-XVI-HD ▶セットでお買い上げの方に●ディスケット10枚●ジョイカード2ケプレゼント中!!

Ⓐセット:CZ-644C-TN+CZ-606D-TN…定価¥597,800▶特価価格はTEL下さい。

12回	39,000	24回	20,700	36回	14,400	48回	11,300	60回	9,500

Ⓑセット:CZ-644C-TN+CZ-614D-TN …定価¥653,000▶特価価格はTEL下さい。

12回	42,600	24回	22,600	36回	15,700	48回	12,400	60回	10,400

※上記のモニターを、CZ-604D(定価¥94,800)、CZ-605D(定価¥115,000)、CU-21HD(定価¥148,000)に変更の場合、TEL下さい。超特価で販売致します。

X68000シリーズ〜P&Aスペシャルセット

(送料¥2,000)

注目!!
「P&Aスペシャルセット」にもれなくプレゼント!!
●上記XVI/XVI-HDのプレゼント
①、②、③+ⒶorⒷのほかに、さらにその上、目にやさしい。
Ⓒ「高性能CRTフィルター(¥19,800)」又は、
Ⓓ「SX-WINDOW. Ver1.1」(¥9,800)
をプレゼント!!

※セットでお買い上げの方に、
●ディスケット10枚
●ジョイカード2個
プレゼント中!!

SUPER

Ⓐセット:P&A特選セット
■CZ-604C (本体定価¥348,000)
⊕
■CZ-606D (モニター定価¥79,800)
P&A超特価 ¥298,000

Ⓑセット
■CZ-604C+CZ-604D
定価¥442,800…▶特価¥300,000
Ⓒセット
■CZ-604C+CZ-607D
定価¥447,800…▶特価¥312,000
Ⓓセット
■CZ-604C+CZ-614D
定価¥483,000…▶特価¥333,000
Ⓔセット
■CZ-604C+CU-21HD
定価¥496,000…▶特価¥340,000

PRO-II

Ⓐセット:P&A特選セット
■CZ-653C (本体定価¥285,000)
⊕
■CZ-606D (モニター定価¥79,800)
P&A超特価 ¥242,000

Ⓑセット
■CZ-653C+CZ-604D
定価¥379,800…▶特価¥247,000
Ⓒセット
■CZ-653C+CZ-607D
定価¥384,800…▶特価TEL下さい。
Ⓓセット
■CZ-653C+CZ-614D
定価¥420,000…▶特価¥279,000
Ⓔセット
■CZ-653C+CU-21HD
定価¥433,000…▶特価¥284,000

SUPER-HD

Ⓐセット:P&A厳選セット
■CZ-623C (本体価格¥498,000)
⊕
■CZ-606D (モニター定価¥79,800)
P&A超特価 ¥376,000

Ⓑセット
■CZ-623C+CZ-604D
定価¥592,800…▶特価¥382,000
Ⓒセット
■CZ-623C+CZ-607D
定価¥597,800…▶特価¥390,000
Ⓓセット
■CZ-623C+CZ-614D
定価¥633,000…▶特価¥415,000
Ⓔセット
■CZ-623C+CU-21HD
定価¥646,000…▶特価¥418,000

EXPERII

Ⓐセット:P&A厳選セット
■CZ-603C (本体価格¥338,000)
⊕
■CZ-606D (モニター定価¥79,800)
P&A超特価 ¥265,000

Ⓑセット
■CZ-603C+CZ-604D
定価¥432,800…▶特価¥270,000
Ⓒセット
■CZ-603C+CZ-607D
定価¥437,800…▶特価¥282,000
Ⓓセット
■CZ-603C+CZ-614D
定価¥473,000…▶特価¥303,000
Ⓔセット
■CZ-603C+CU-21HD
定価¥486,000…▶特価¥310,000

●本広告の掲載の商品の価格については、消費税は含まれておりません。●営業時間=平日AM10:00〜PM7:00、日祭AM10:00〜PM6:00

回～84回払いまでOK.!!

★頭金なし!★即日発送

●価格は流通事情により変動致しますので、銀行振込・書留等の送付前に、あらかじめお電話にてご確認下さい。

P&Aがズバリ超特価セールでご奉仕!!

立寄り下さい。専門係員が説明いたします。
価で受付します。詳しくは電話にてお問合せ下さい。
価の20%引きOK! TELください。

全国通販

超特価でクレジットが組める!!

X68000用ソフトコーナー(送料1ヶ～5ヶまで¥500)

商品	定価	特価
Z's STAFF PRO68K Ver.2.0(ツァイト)	¥58,000	¥38,000
Z's TRIPHONY デジタルクラフト(ツァイト)	¥39,800	¥27,800
テラッツォ(ハミングバード)	¥19,400	¥14,200
KAMIKAZE(サムシング・グッド)	¥68,000	¥44,800
C & Professional Pack(マイクロウェアジャパン)	¥58,000	¥41,000
Final Ver3.2(エーエスピー)	¥38,000	¥29,600
C-compiler PRO68K Ver.2 CZ-245L	¥44,800	¥33,300
CARD PRO68K CZ226BS	¥29,800	¥21,200
¥BAS to C CHECKER CZ-260LS	¥9,800	¥7,400
OS-9/X68000 CZ219SS	¥29,800	¥22,500
AI-68K CZ234LS	¥188,000	¥138,000
THE 福袋 V2.0 CZ224LS	¥9,900	¥7,400
SOUND PRO68K CZ-214MS	¥15,800	¥11,400
MUSIC PRO68K CZ213MS	¥18,800	¥13,400
Sampling PRO68K CD215MS	¥17,800	¥12,700
MUSIC-studio PRO68K CZ-252MS	¥15,800	¥21,400
MUSIC-PRO68K〔MIDI〕247MS	¥28,800	¥20,700
New-print Shop 221HS	¥19,800	¥15,500
Communication 223CS	¥19,800	¥14,200
Communication Ver.2 CZ-257CS	¥19,800	¥15,500
C-TRACE68 Ver.3.0(キャスト)	¥98,000	¥69,000
サイクロンEXPRESSα68	¥98,000	¥69,800
G68K Ver2 PRO	¥22,000	¥17,500
SX-WINDOW CZ-259SS	¥6,800	¥4,900
Gツール(ザインソフト)	¥28,000	¥18,900
たーみのる2(SPS)	¥17,800	¥13,300
マジックパレット(ミュージカルプラン)	¥19,800	¥14,500
Hyper word CZ-251BS	¥39,800	¥29,600

●ゲームソフト20%OFF OK!!(一部ソフト除く)

周辺機器コーナー(送料¥500)

No.	商品	定価	特価
1	CZ-8NSI	¥188,000	¥140,000
2	CZ-6VTI	¥69,800	¥52,500
3	CZ-6TU	¥33,100	¥24,500
4	BF-68PRO	¥19,800	¥15,300
5	CZ-6BEI	¥35,000	¥26,000
6	CZ-6BEIA	¥38,000	¥28,600
7	CZ-6BE2A	¥59,800	¥44,200
8	CZ-6BE2B	¥54,800	¥40,800
9	CZ-6BFI	¥49,800	¥38,200
10	CZ-6BPI	¥79,800	¥60,000
11	CZ-6BMI	¥26,800	¥20,300
12	CZ-6EBI	¥88,000	¥66,500
13	AN-S100	¥36,600	¥28,500
14	CZ-6SDI	¥44,800	¥35,000
15	CZ-6BN1	¥29,800	¥22,600
16	CZ-6BV1	¥21,000	¥15,900
17	CZ-64H	¥120,000	¥91,500
18	CZ-6BG1	¥59,800	¥45,000
19	CZ-6BU1	¥39,800	¥30,300
20	CZ-6PVI	¥198,000	¥153,000
21	CZ-6BS1	¥29,800	¥22,300
22	CZ-8NJ2	¥23,800	¥18,500
23	CZ-6BL2	¥298,000	¥222,000
24	JX-100S	¥89,800	¥48,500
25	JX-220X	¥168,000	¥126,000
26	IO-735XB	¥248,000	¥169,000

(IO-735XBご購入の方「BANANA-PRINT」プレゼント!!

X68000用ハードディスク(送料¥1,000)

アイテック(SCSIタイプ)

■TX-130(130MB)…定価¥138,000▶特価¥ 98,000
(送料・消費税込み¥101,970)

■TX-180(180MB)…定価¥185,000▶特価¥132,000
(送料・消費税込み¥136,990)

プリンター(ケーブル・用紙付) (送料¥1,000)

■CZ-8PC5-BK NEW ……定価¥96,800▶特価価格はTEL.!!
■CZ-8PK10 ……定価¥97,800▶特価¥71,000
■CZ-8PG2 ……定価¥160,000▶特価価格はTEL.!!
■CZ-8PG1 ……定価¥130,000▶特価価格はTEL.!!

モデムコーナー (送料¥1,000)

■COMSTARZ CLUB24/5
(NEC)定価¥39,800
特価¥26,500 (送料・消費税込み ¥28,325)

■MD-24FB5V
(オムロン)定価¥39,800
特価¥27,400 (送料・消費税込み ¥29,252)

P&A特選パソコンラック (送料無料)

①3段¥8,800 ②4段¥9,800 ③5段¥13,800

全機種=移動自由(キャスター付)・キーボード収納可(5段のみ)=1230(H)×600(D)×650(W)

中古パソコンはP&Aにおまかせ!!

その場で高価現金買取り・高価下取りOK!!

■まずはお電話下さい。
03-3651-1884、FAX:03-3651-0141

■下取り・買取りでお急ぎの方、直接当社に来店、または、宅急便にてお送り下さい。

●下取りの場合……………価格は常に変動していますので査定額をお電話で確認して下さい。(差額は、P&A超低金利クレジットをご利用下さい。)

●買取りの場合……………現品が着き次第、2日以内に買取り金額を連絡し、振込み、又は書留でお送り致します。

●近郊の方は、P&A本店まで、直接お持ち下さい。即金にて、¥1,000,000までお支払い致します。

《便利な超低金利クレジットをご利用下さい》

●月々¥1,000円からOK!! ●ボーナス払いOK(夏冬10回までOK)
●支払い回数 1回～84回 ●お支払いは、8ヶ月先からでもOK!!

アフターサービス万全
全商品保証付。専門の担当者がお客様の立場で対応します。
初期不良、輸送トラブルetc.
万が一初期不良、輸送トラブルが発生しました際には、即交換させていただきます。

●定休日/毎週水曜日=第3水曜(祭日の場合は翌日になります)

通信販売お申し込みのご案内

〔現金一括でお申し込みの方〕
●商品名およびお客様の住所・氏名・電話番号をご記入の上、代金を当社まで、現金書留でお送りください。(プリンター・フロッピーの場合、本体使用機種名を明記のこと)

〔銀行振込でお申し込みの方〕
●銀行振込ご希望の方は必ずお振込みの前にお電話にてお客様のご住所・お名前・商品名等をお知らせください。
(電信扱いでお振込み下さい。)

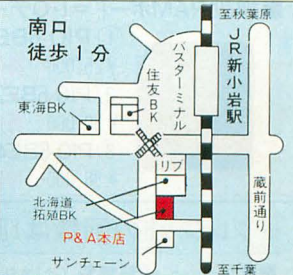
〔振込先〕 住友銀行 新小岩支店
普通預金 1451576 ㈱ピー・アンド・エー

〔クレジットでお申し込みの方〕
●電話にてお申し込みください。クレジット申し込み用紙をお送りいたしますので、ご記入の上、当社までお送りください。
●現金特別価格でクレジットが利用できます。残金のみに金利がかかります。
●1回～84回払いまで出来ます。但し、1回のお支払い額は¥1000円以上。

超低金利クレジット率

回数	3	6	10	12	18	24	36	48	60	72	84
手数料	3.5	4.5	6.0	6.0	11.0	12.5	17.5	23.0	29.5	38.0	45.5

マイコン専門ショップ
P&A
株式会社ピー・アンド・エー
〒124 東京都葛飾区新小岩2丁目1番地19号
☎03-3651-0148(代) FAX.03-3651-0141

営業時間
平日:AM10:00～PM7:00
日祭:AM10:00～PM6:00

●現金書留及び銀行振込でお申し込みの方は、上記商品の料金に3%加算の上でお申し込み下さい。詳しくは、お電話でお問い合せ下さい。

■店頭にて、新作ゲームソフト25〜30%OFF!!（税別）、超低金利オクトハッピークレジットをご利用下さい!!

──オクトで始まるパソコンワールド──

☎03-3730-6271

●営業時間 AM 11:00〜9:00/日曜・祭日PM7:00　電話一本で、ハイ即納

〒144 東京都大田区蒲田4-6-7 FAX03-3730-6273

全国通販

●定休日毎週火曜日 祭日の場合翌日になります。

オクト ラクラククレジット

3回	3.5	6回	4.5	10回	6.0	12回	6.0	15回	9.0	18回	11.0
20回	12.0	24回	12.5	30回	17.0	36回	17.5	48回	23.0	60回	33.0

OCT-1 システム インフォメーション

▶全商品保証付（メーカー保証）
▶超低金利ハッピークレジット（1回〜60回）頭金ナシOK!
▶ボーナス一括払いOK! ボーナス2回払いOK!!
▶配達日の指定OK!（万全なサポート体制）
▶商品の組合せ自由! オクトフリーダムシステム
▶店頭デモンストレーション実施中

オクト セレクテッドシステム

広告掲載商品以外の製品も取扱っております。

蒲田

翌月末一括（9月末）払いOK!! 手数料無料!!
サマーフェスティバル! それは夏の市大事件!!
あなたもTELしてこの感動を‥!! NOW ON SALE

■朗報デス。翌月末一括（9月末）払いOK!!手数料無料。ご利用下さい。■店頭にて、新作ゲームソフト25〜30%OFF!!

■便利です。夜9時まで営業しております。お立寄り下さい。お待ちしております!!

X68000 XVI エクシヴィ

■16MHz■
■SX-WINDOW ver1.1■
■Attachment MEMORY BORD■

快速16MHz 鮮烈デビュー!!

■CZ-634C-TN（定価￥368,000）

Ⓐ●CZ-634C-TN
●CZ-614D-TN NEW
定価合計￥503,000

12回	￥34,000	24回	￥18,000	36回	￥12,500	48回	￥9,800

Ⓑ●CZ-634C-TN
●CZ-607D-TN NEW
定価合計￥467,800

12回	￥31,700	24回	￥16,800	36回	￥11,700	48回	￥9,100

Ⓒ●CZ-634C-TN
●CZ-606D-TN
定価合計￥447,800

12回	￥30,400	24回	￥16,100	36回	￥11,200	48回	￥8,800

■CZ-644C-TN（定価￥518,000）

Ⓓ●CZ-644C-TN
●CZ-614D-TN NEW
定価合計￥653,000

12回	￥43,500	24回	￥23,100	36回	￥16,000	48回	￥12,600

Ⓔ●CZ-644C-TN
●CZ-607D-TN NEW
定価合計￥617,800

12回	￥41,300	24回	￥21,900	36回	￥15,200	48回	￥11,900

Ⓕ●CZ-644C-TN
●CZ-606D-TN
定価合計￥597,800

12回	￥40,000	24回	￥21,200	36回	￥14,700	48回	￥11,600

X68000XVI 残暑ぶっとびプレゼント!!

―あなたのオクトから素敵な贈物―

今、XVIをお買い上げいただいた方は、プレゼントの①番か②番のどちらかお選び下さい。プラス③番はもれなくプレゼント!!

① 遥かなるオーガスタ ゴルフゲームの決定版（定価￥12,800）＋ 大人気 大戦略II（キャンペーン版）不朽の名作X68000版（定価￥9,800）

or

② インテリジェントコントローラ ■CZ-8NJ2（CYBER STICK）シューティングゲーマーの必須アイテム!!（定価￥23,800）

③ MD-2HD（10枚）シリコンキーボードカバー もれなく!! サービス!!

▶現金超特価 ￥TEL下さい!!▶

※どちらかお選び下さい!!（どっちが得かヨーク考えてネ!）

特選周辺機器（送料￥500）

●SX-68M MIDIインターフェースボード（システムサコム）￥19,800…特価￥14,000
●Fine Scanner X68（HAL研究所）（HGS-68）￥39,800…特価￥25,800

■増設RAMボード＝I・Oデータ

① PIO-6BE1-A（1MB）￥25,000…特価￥16,500
② PIO-6BE2-2M（2MB）￥50,000…特価￥32,500
③ PIO-6BE4-4M（4MB）￥88,000…特価￥56,000

周辺機器コーナー（送料￥500）

●CZ-6BEI IBM増設RAMボード……（￥35,000）▶特価￥26,000
●CZ-6BEIB IBM増設RAMボード……（￥28,000）▶特価￥21,000
●CZ-6BE2 2MB増設RAMボード……（￥79,800）▶特価￥60,000
●CZ-6BE4 4MB増設RAMボード……（￥138,000）▶特価￥103,000
●CZ-6BFI 増設用RS-232Cボード……（￥49,800）▶特価￥38,000
●CZ-6BGI GP-IBボード……（￥59,800）▶特価￥45,000
●CZ-6BMI MDIボード……（￥26,800）▶特価￥20,000
●CZ-6BNI スキャナ用パラレルボード……（￥29,800）▶特価￥22,500
●CZ-6BPI 数値演算プロセッサボード……（￥79,800）▶特価￥60,000
●CZ-6BOI ユニバーサルI/Oボード……（￥39,800）▶特価￥30,500
●CZ-6EBI/BK 拡張I/Oボックス……（￥88,000）▶特価￥65,800
●CZ-6VTI/BK カラーイメージ・ユニット……（￥69,800）▶特価￥52,000
●CZ-8NM2A マウス……（￥6,800）▶特価￥5,300
●CZ-8NTI マウストラックボール……（￥9,800）▶特価￥7,500
●CZ-8NSI カラーイメージスキャナ……（￥188,000）▶特価￥137,000
●CZ-6BCI FAXボード……（￥79,800）▶特価￥60,500
●CZ-8TM2 モデムユニット……（￥49,800）▶特価￥38,000
●CZ-64H 増設ハードディスク……（￥120,000）▶大特価
●CZ-6TU GY/BK RGBシステムチューナー……（￥33,100）▶特価￥24,500
●BF-68PRO 高性能CRTフィルター……（￥19,800）▶特価￥15,000
●CZ-6MOI 光磁気ディスクユニット……（￥450,000）▶特価￥328,000
●CZ-6BSI SCSIインターフェースボード……（￥29,800）▶特価￥22,200
●CZ-6BL2 LANボード……（￥298,800）▶特価￥220,000
●CZ-6BVI（ビデオボード）……（￥21,000）▶特価￥15,500
●CZ-6BE2A 2MB増設RAMボード……（￥59,800）▶特価￥43,800
●CZ-6BE2B 2MB増設メモリ（チップ型）……（￥54,800）▶特価￥40,000
●CZ-6BP2 数値演算プロセッサ……（￥45,800）▶特価￥34,000
●AN-S100 スピーカーシステム（2本1組）……（￥36,600）▶特価￥27,000

※クレジットの回数は1回〜60回、ボーナス併用などありますのでお電話でお問合せ下さい。

■本体セット：送料無料（注）本体セット以外の周辺機器（プリンター、モデム、HDD等）及びソフトの送料は、北海道・九州地区＝1ケロ￥1500、■その他離島地区は、1ケロ￥2000となります。
※上記料金には、消費税は含まれておりません。消費税が付加されますので、詳しくは、電話でお問合せ下さい。

■特に人気のある商品によっては、しばらくお待ち願うことがありますのでご了承下さい!!

■超低金利クレジットをご利用下さい。1回〜60回払い、頭金ナシ!!ボーナス1回及び2回払いOKです。

■ビッグバーゲンセール実施中!!ゲームソフト(ビジネス)新製品続々入荷中!!

X68000 ラストチャンス!!

■SUPER／PROII／SUPER-HD

◀大人気!!＊大戦略II シュミレーションゲーム プレゼント (定価￥9,800)
★JOY CARD (連射式)×2個
★MD-2HD 10枚

限定

■SUPER(定価￥348,000) CZ-604C-TN

■PROII(定価￥285,000) CZ-653C-BK/GY

■SUPER-HD(定価￥498,000) CZ-623C-TN

15型カラーディスプレイTV

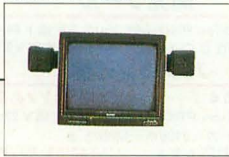
CZ-614D-TN 定価￥135,000

14型カラーディスプレー

CZ-606D(GY/BK/TN) 定価￥79,800

21型カラーディスプレイ

CU-21HD 定価￥148,000

CZ-8NJ2 限定
●インテリジェントコントローラ
定価￥23,800
超特価￥18,000

(送料・消費税込)

Ⓐ CZ-604C＋CZ-614D……定価合計￥483,000▶￥338,000

12回	￥29,800	24回	￥15,800	36回	￥11,000	48回	￥ 8,600	60回	￥ 7,400

Ⓑ CZ-653C＋CZ-614D……定価合計￥420,000▶￥大 特 価

12回	?	24回	?	36回	?	48回	?	60回	?

Ⓒ CZ-623C＋CZ-614D……定価合計￥633,000▶￥418,000

12回	￥36,900	24回	￥19,500	36回	￥13,600	48回	￥10,700	60回	￥ 9,200

Ⓓ CZ-604C＋CZ-606D……定価合計￥427,800▶￥298,000

12回	￥26,300	24回	￥13,900	36回	￥ 9,700	48回	￥ 7,600	60回	￥ 6,600

Ⓔ CZ-653C＋CZ-606D……定価合計￥364,800▶￥大 特 価

12回	?	24回	?	36回	?	48回	?	60回	?

Ⓕ CZ-623C＋CZ-606D……定価合計￥577,800▶￥389,000

12回	￥34,300	24回	￥18,200	36回	￥12,600	48回	￥ 9,900	60回	￥ 8,600

Ⓖ CZ-604C＋CU-21HD……定価合計￥496,000▶￥346,000

12回	￥30,500	24回	￥16,200	36回	￥11,200	48回	￥ 8,800	60回	￥ 7,600

Ⓗ CZ-653C＋CU-21HD……定価合計￥433,000▶大 特 価

12回	?	24回	?	36回	?	48回	?	60回	?

Ⓘ CZ-623C＋CU-21HD……定価合計￥646,000▶￥430,000

12回	￥37,900	24回	￥20,100	36回	￥14,000	48回	￥11,000	60回	￥ 9,500

★本体セットは、1ヶ月間だけの大特価セール!!
★クレジット価格は、消費税込みですョ。ご利用下さい!!

X68000ソフト大セール実施中!!(ゲームソフト25〜30%OFF) (送料￥500)

〈グラフィック〉●Z's STAFF PRO68K Ver.2.0 (シャフト)定価￥58,000 ……特価￥38,500

〈開発ツール〉●C-コンパラPRO68KV.2 定価￥44,800 CZ-245IS ……特価￥33,000

〈データベース〉●CARD PRO68K Ver.2.0 定価￥29,800 CZ-253BS ……特価￥21,000

〈グラフィック〉●C-TRACE 68 Ver.3.0 定価￥98,000 ……特価￥69,000

〈C言語〉●C & Professional Pack 定価￥58,000 ……特価￥40,500

〈音楽〉●Music studio PRO68K Ver. 2.0 定価￥28,800 CZ-261MS ……特価￥21,300

〈CGシール〉●CANVAS PRO68K 定価￥29,800 CZ-249GS ……特価￥22,200

〈ワープロ〉●Multiword PRO68K 定価￥32,000 CZ-225BS ……特価￥23,800

〈通信〉●Tlepotion PRO68K 定価￥22,800 CZ-258BS ……特価￥17,000

型名	商品	定価	特価
CZ-212BS	BUSINESS PRO-68K	￥ 68,000	￥ 48,000
CZ-213MS	MUSIC PRO68K	￥ 18,800	￥ 13,500
CZ-214MS	SOUND PRO-68K	￥ 15,800	￥ 11,500
CZ-215MS	Sampling PRO-68K	￥ 17,800	￥ 12,800
CZ-219SS	OS-9/X68000	￥ 29,800	￥ 21,000
CZ-220BS	DATA PRO-68K	￥ 58,000	￥ 41,000
CZ-223CS	Communication PRO-68K	￥ 19,800	￥ 14,300
CZ-224LS	THE 福袋 V2.0	￥ 9,900	￥ 7,500
CZ-241BS	システム手帳リフィル集	￥ 9,800	￥ 7,500
CZ-242BS	活用フォーム集	￥ 9,800	￥ 7,500
CZ-244SS	Homan 68K Ver2.0	￥ 9,800	￥ 7,500
CZ-247MS	MUSIC PRO-68K(MIDI)	￥ 28,800	￥ 20,800
CZ-240BS	Stationery PRO-68K	￥ 14,800	￥ 11,500
CZ-243BS	CYBER NOTE PRO-68K	￥ 19,800	￥ 15,200

型名	商品	定価	特価
	Z's TRIPHNY(デジタルクラフト)	￥ 39,800	￥ 27,500
	テラッツオ(ハミングバード)	￥ 19,400	￥ 14,000
	KAMIKAZE(サムシンググッド)	￥ 68,000	￥ 44,500
	Final Ver.3.2(エーエスピー)	￥ 38,000	￥ 29,500
	サイクロンEXPRESSα68	￥ 98,000	￥ 69,500
	Gツール(ザインソフト)	￥ 28,000	￥ 18,800
	たーみのる2(SPS)	￥ 17,800	￥ 13,200
	G68K Ver.2 PRO	￥ 22,000	￥ 17,500
CZ-259SS	SX-WINDOW Ver.1.0	￥ 6,800	￥ 5,000
CZ-251BS	ハイパーワード	￥ 39,800	￥ 29,600
CZ-260LS	XBAS to CHECKER PRO68K	￥ 9,800	￥ 7,500
CZ-234LS	AI-68K	￥188,000	￥139,000
CZ-255GS	CANVASドローグラフィックLIB	￥ 8,800	￥ 6,600
CZ-256GS	CANVASドローグラフィックVol.2	￥ 8,800	￥ 6,600

熱転写カラー漢字プリンター (送料￥1,000)

■CZ-8PC5

●48ドット
●熱転写カラー漢字プリンター
定価￥96,800
特価￥TEL下さい!!(ケーブル用紙付)

ハードディスク (送料￥1,000)

■アイテック

X68000用ハードディスク

●TX-80(定価￥108,000)…▶大特価￥79,000 (80MB、SCSI、SASI両対応)
●TX-130(定価￥138,000)…▶大特価￥99,000 (130MB、SCSI対応)
●TX-180(定価￥185,000)…▶大特価￥134,000 (180MB、SCSI対応)

夏休み限定フェア パソコンラック〈送料無料〉

Ⓐ5段キャスター付
スライド式キーボード台
●1150(H)×640(W)×600(D)
定価￥38,000
特価 (￥15,000) ￥14,000

Ⓑ4段キャスター付
●1250(H)×640(W)×700(D)
定価￥29,800
特価 (￥11,000) ￥10,000

店頭新作ゲームソフト25〜30%OFF!!ビジネスソフト25%より特価中

★通信販売お申込みのご案内★ 〒144 東京都大田区蒲田4-6-7 TEL:03-3730-6271

お申込みはお電話でお願いします。お客様の〈住所〉〈氏名〉〈電話番号〉及び〈商品名〉をお知らせ下さい。●入金確認後ただちに商品をご送付いたします。

現金一括払い
銀行振込:お近くの銀行より(電信扱い)にてお振込み下さい。
現金書留:封筒の中に住所・氏名・商品名をご記入の上当社までお送り下さい。

クレジット
専用お申込用紙をお送り致します。ので、必要事項をご記入、ご捺印の上ご返送下さい。手続きは簡単です。

オクト ラクラク クレジット表

3回	3.5	6回	4.5	10回	6.0	12回	6.0
15回	9.0	18回	11.0	20回	12.0	24回	12.5
30回	17.0	36回	17.5	48回	23.0	60回	33.0

振込先
富士銀行 久ヶ原支店 (当)No.1824
三菱銀行 蒲田支店 (当)No.0278691
株式会社 億人(オクト)

※掲載の価格は変動しますので、まずは、お電話にてご確認ください。
※上記料金には、消費税は含まれておりません。消費税が付加されますので、詳しくは電話でお問合せ下さい。
※銀行振込、または、現金書留でご注文の際には、あらかじめ電話でご確認の上、お申し込み下さい。

信頼と安さのTSUKUMO

掲載商品2万円以上送料無料（離島を除く）

Tech(テク) Tech(テク) 行こう ツクモの日

9月9日 9/9(月)〜9/19(木)迄

ツクモの日記念セール！

ツクモパソコン本店2FはX68000ワールド

X68000 XVI 快速16MHz

- CPUクロック周波数スピードアップ（16MHz）
- 増設メモリ本体内蔵可能（8MBまで）
- NEW SX-WINDOW搭載

■X68000XVI(CZ-634C-TN)
標準タイプ………定価￥368,000

■X68000XVI-HD(CZ-644C-TN)
HD内蔵タイプ……定価￥518,000

（買い換え・下取りも取り扱っております 是非、お尋ね下さい）

X68000XVI+HD「Power User」セット

- CZ-634C-TN（X68000XVI本体）……￥368,000
- CZ-614D-TN（専用高解像度ディスプレイ）……￥135,000
- TX-180B（180MB SCSIハードディスク）……￥185,000
- CZ-6BE2A（2MB増設RAMボード）……￥59,800

合計定価￥747,800

記念特価 ￥618,000（消費税別途￥18,540）
クレジット例（60回払・税込）
初回￥19,742＋月々￥14,500×59回

買い換え・下取りも行なっております。詳しくは下記へ

「ツクモの日」特別セット 期間限定

ワープロソフトセット

- Multiword（シャープ）ワープロソフト……￥32,000
- HGS-68（HAL研）ハンディスキャナー……￥39,800

合計定価￥71,800

記念特価￥56,000（消費税別途￥1,680）

CGツールセット

- IO-735-X（カラーイメージジェットプリンタ）……￥248,000
- JX-220-X（A4版カラーイメージスキャナー）……￥168,000
- X68000用プリンターケーブル……￥4,800
- Z's STAFF PRO-68K Ver2.0……￥58,000

合計定価￥478,800

記念特価￥368,000（消費税別途￥11,040）
クレジット例（36回払・税込）
初回￥14,719＋月々￥12,900×35回

9/9(月)〜9/19(木)ツクモの日記念 ディスケットプレゼント

日頃のお客様のご愛顧に感謝して……
期間中にツクモTSシリーズのTSドライブをお買上げのお客様にディスケット3.5インチ＝5枚or5インチ＝10枚をプレゼント!!

ツクモオリジナルドライブ

X68000シリーズ専用 3.5インチフロッピーディスクドライブ
大好評につき在庫お問合せ下さい

TS-3XR1 定価￥44,800

仕様
- 1ドライブタイプ
- 3.5インチ2DD／2HD対応ドライブ使用
- 2DD用デバイスドライバ付属

※SX-WINDOWは対応しておりません

発売記念特価￥35,800（消費税別途￥1,074）

安心 迅速 高額

買い取りのツクモニューセンター店

ツクモ買い取りセンター
—好評買い取り中！—

電話受付(03)3251-9977（AM11:00〜PM5:00迄）
FAX受付(03)3251-0299（24時間）

ツクモグローバルカード

大人気／入会者募集中／
国内・外で活躍／ 使って便利、持ってて安心／ツクモグローバルカードはジャックス・VISAとの提携カードです。ツクモ各店でのお買物がらくらくできるうえに、国内はもとより海外での分割ショッピングもOK／
お申し込みは☎03(3251)9898 又は店頭にて／
18才以上なら学生でもOK／

※各店頭では、JCB・日本信販・DC・セントラル・マスター他各種カードも取り扱っております。

X68000用ハードディスク

—大容量記憶装置—

80MB SCSI/SASI両対応タイプ
TX-80 定価￥108,000
特価￥88,000（消費税別途￥2,640）

130MB SCSI対応タイプ
TX-130 定価￥138,000
特価￥110,000（消費税別途￥3,300）

180MB SCSI対応タイプ
TX-180
定価￥185,000
特価￥148,000（消費税別途￥4,440）

SCSIタイプHDDの場合、本体がSUPER/XVI以外の場合にはSCSIボード(CZ-6BS1)が必要です。

X68000用増設メモリーボード

1MB増設RAMボード（ACE/PRO/PRO2シリーズ用）特価￥17,500
2MB増設RAMボード…特価￥34,800
4MB増設RAMボード…特価￥61,500

※計測技研のメモリーボードも取扱っておりますので、価格についてはお尋ね下さい。

開発ツール

- ■C Compiler PRO-68K Ver2.0定価￥44,800
- ■XBAS TO C CHECKER PRO-68K定価￥9,800

パソコン通信

- ■た〜みのる 2……ツクモ特価￥14,200
- ■Telepotion PRO-68K……定価￥22,800
- ■一流メーカー2400ボーMNP5 & V42bis対応モデム……ツクモ特価￥31,800

電子手帳

- ■ハイパー電子システム手帳 PA-9500……定価￥48,000 ツクモ特価￥43,000
- PA-9550……定価￥59,000
- ■スタイリッシュ電子システム手帳 PA-X1……定価￥29,800 ツクモ特価￥26,000
- ■Telepotion PRO68K……定価￥22,800
- ■CE-300L電子手帳通信ケーブル……ツクモ特価￥2,380

※ポケットコンピュータも取扱っております。価格はお尋ね下さい。

ビジネスツール

- ■Hyper WORD……定価￥39,800
- ■Multiword NEW……定価￥32,000
- ■FIXER Ver4.0……ツクモ特価￥15,800
- ■CARD PRO-68K Ver2.0 NEW 定価￥29,800

アートツール（ハード）

- ■JX-220X A4サイズカラーイメージスキャナー……定価￥168,000
- ■HGS-68 ファインスキャナーX68……ツクモ特価￥31,800
- ■CZ-6VT1 カラーイメージユニット 定価￥69,800
- ■CZ-6BV1 ビデオボード……定価￥21,000
- ■XAV-1Sアナログ RGB・S端子変換ユニット ツクモ特価￥6,890
- ■CZ-8PC5 48ドットカラー漢字熱転写プリンター NEW……定価￥96,800

アートツール（ソフト）

- ■CANVAS PRO-68K……定価￥29,800
- ■Easy Paint SX-68K(CZ-263GW) 定価￥12,800
- ■Z's STAFF PRO-68K Ver2……ツクモ特価￥46,400
- ■マジックパレット……ツクモ特価￥15,800

冬のボーナス一括払い受付中!! 詳しくは☎03(3251)9911へ

X68000 ユーザー必須のコンピュータミュージック特別セット

Aセット
- CM-32L……￥69,000
- SX-68M……￥19,800
- Musicstudio Mu-1 Ver1.4……￥19,800

合計定価￥108,600
ツクモ特価￥88,000（消費税別途￥2,640）
クレジット例（18回払・税込）
初回￥7,223＋月々￥5,600×17回

Bセット
- CM-64……￥129,000
- SX-68M……￥19,800
- Musicstudio Mu-1 Ver1.4……￥19,800

合計定価￥168,600
ツクモ特価￥138,000（消費税別途￥4,140）
クレジット例（24回払・税込）
初回￥7,603＋月々￥6,900×23回

マニアセット NEWセット
- SC-55（ローランドサウンドキャンバス）……￥69,000
- SX-68M……￥19,800
- Mu-1 SUPER……￥39,800

合計定価￥128,600
ツクモ特価￥99,000（消費税別途￥2,970）
クレジット例（10回払・税込）
初回￥11,517＋月々￥10,900×9回

SUPERマニアセット NEWセット
- CM-64……￥129,000
- SX-68M……￥19,800
- Mu-1 SUPER……￥39,800

合計定価￥188,600
ツクモ特価￥154,000（消費税別途￥4,620）
クレジット例（18回払・税込）
初回￥10,940＋月々￥9,900×17回

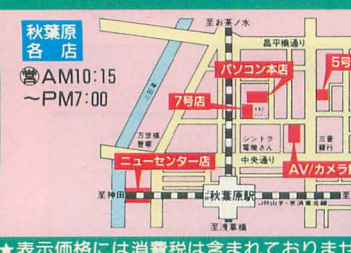
Aセット と Bセット の場合、「Musicstudio PRO68K Ver2.0」又は、「Music PRO68K <MIDI>」のソフトの場合には￥9,500プラスになります。また、これらのソフトウェアがバージョンアップにより価格が変更になった場合には変更となります。

ローランド 追加オプション機器
- ステレオマイクロモニター CS-10……定価￥17,000
- MIDIキーボードコントローラー PC-200……定価￥36,000
- はなうたくん CP-40……定価￥33,000

7号店2F（MIDI専門フロア）にもございます。☎03(3253)4199 担当・佐々木

商品のご注文は通販受注専用センター フリーダイヤル **0120-377-999**

商品についての詳しいお問い合わせは各店、又は☎03(3251)9911へ

秋葉原各店
営AM10:15〜PM7:00

ツクモは「スーパーX PRO SHOP」です。

九十九電機株
〒101-91 東京都千代田区神田郵便局私書箱135号

ツクモパソコン本店2F ☎03-3253-5599（担当／荒井）㊡毎週木曜と8/14

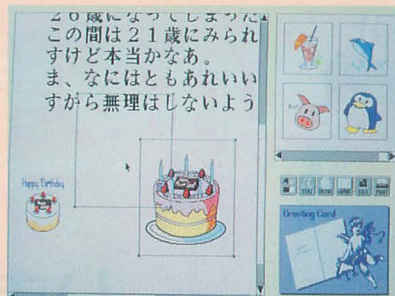
便利で安心な通信販売
ツクモ通販センター☎03-3251-9911

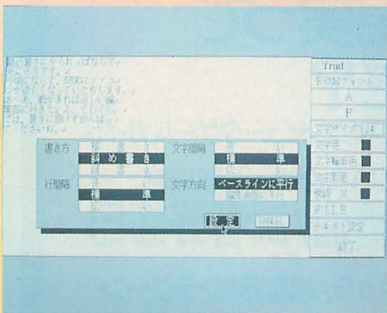
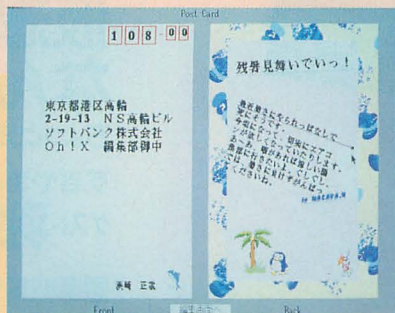
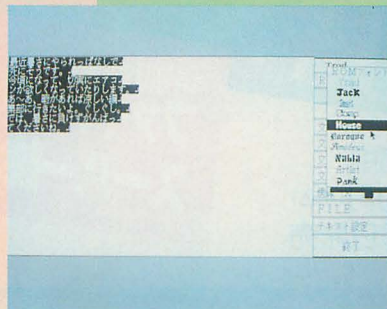
- ■ツクモニューセンター店 ☎03-3251-0987（担当／福地）㊡毎週木曜と8/14
- ■ツクモAV/カメラ館B1 ☎03-3254-3999（担当／川名）㊡毎週水曜と8/15
- ■ツクモ5号店 ☎03-3251-0531（担当／森）㊡毎週木曜と8/14
- ■名古屋1号店 ☎052-263-1655（担当／吉高）㊡毎週月曜と8/19〜21
- ■名古屋2号店 ☎052-251-3399（担当／横山）㊡毎週水曜と8/21〜23
- ■ツクモ札幌店 ☎011-241-2299（担当／田口）㊡毎週木曜と8/7

★表示価格には消費税は含まれておりません。
★商品のご注文は在庫確認の上お願いします。

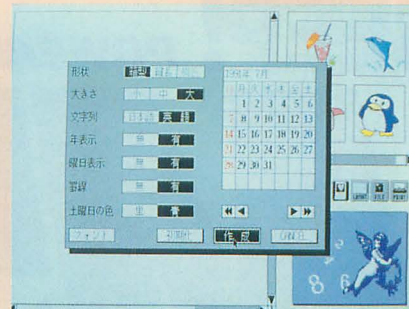
カード払い	全国代金引き換え配達	クレジット払い	現金書留払い	銀行振込払い	各種リース払い
通信販売での御利用カード、ツクモグローバルカード、VIPカード、セントラル、ジャックス※御本人様より電話で通信販売部へお申し込み下さい。	お申し込みは☎03-3251-9911へお電話1本！配達日の指定もできます。	月々￥3,000以上の均等払いも頭金なし、夏・冬ボーナス2回払いも受付中！	〒101-91東京都千代田区神田郵便局私書箱135号 ツクモ通販センター Oh!X係	事前に☎でお届け先をご連絡下さい。三和銀行 秋葉原支店（普）1009939 ツクモデンキ	くわしくは各店にお問い合せ下さい。ケースに合わせてご相談にのります！

NEW Print Shop PRO-68K ver2.0

タイトル画面の右下，新編集機能に注目

グラフィックパーツの移動もらくちん

レイアウト画面で全体の位置調整をする

文字の書き方に斜め書きが追加された

文字を右下がりに指定して配置したところ

テキストエディタで文字フォントも簡単に指定できる

色属性を変更してモノクロ／カラー表示を切り換えられる

グラフィックエディタで回転機能を使う

新しく加わったカレンダー編集機能

オリジナルの印刷物を編集・作成するためのツール「NEW Print Shop」がバージョンアップされました。

オリジナルカードの作成手順はいたって簡単。画面にグラフィックを配置，テキストを編集して必要ならそれぞれの文字に装飾を施し，それらをレイアウト画面で自由に動かすだけです。面倒な操作は一切必要なく，マウス片手に「ここにこいつを持ってきて，こいつをあっちに……」と自分の思いどおりに作業することができます。

標準で付属しているデータも豊富にあってわざわざ自分でデータを作る必要もありません。とはいうものの，やはりオリジナルというからには自分でデータを作成したいもの。そんなときにはちゃんとグラフィックエディタも付属していますし，CANVAS PRO-68K,Z'sSTAFF PRO-68Kのデータを使用することもできます。

文字フォントについても半角文字なら標準で11パターンあり，気分しだいで使い分けることができます。残念ながら全角文字で使用できるのはROMフォントだけですがスムージング処理を加えてあり，それなりに見栄えがします。また，Zeitの書体倶楽部に対応しているので，持っていれば美しいアウトラインフォントを使い，さらにきれいな印刷が可能となっています。

ver2.0で作成できるものは便箋，封筒，ディスクエンベロープ，グリーティングカード，ポスター，横断幕，新たにカセットレーベル，カレンダー編集機能がつき，さらに使用用途が広がりました。

基本的にカラー印刷を目的としていますがモノクロデータの加工・出力もサポートされています。テキストデータも自由な方向に設定可能となり，斜め方向の位置調整も簡単に実現できるようになりました。

ver1.0と比べてみると，それぞれの機能が充実していて全体的にかなり遊べるものに仕上がっています。そして，全体的な処理速度の向上は特にうれしいですね。10月号ではさらに詳しくレビューしますからお楽しみに。

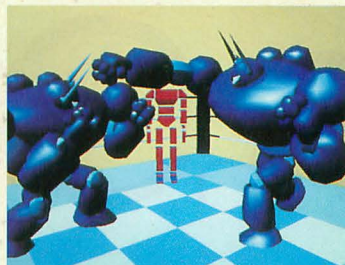
シャープ　20,000円（税別）

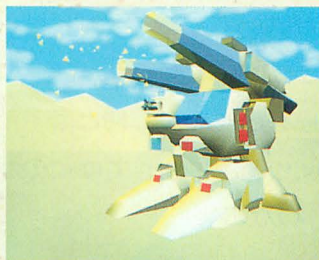
NEW PrintShop PRO-68Kのver.1.0のユーザーの方で，バージョンアップを希望する方は，データディスクのラベル部分を切り取り，住所，氏名，電話番号を明記の上，以下の宛先にお送りください。折り返し，案内状が送られてくるとのことです。
〒351 埼玉県朝霞市浜崎577　東京電化（株）営業部シャープ担当

DōGA・CGアニメーション講座

前回に引き続き，人体モデルのお話です。
今月は前回掲載したBOX君に代わって，悪役ロボットをタイルの上で歩かせてみました。ほかに，ロボットのシャドウボクシングや，実際に戦っている画像を掲載しておきます。右下にあるBOX君のボクシングのモーションデザインと見比べて動きの研究をしてみてください。GALちゃんは，次回こそは暴れ回ってもらいましょう。
ほかに今月は，はやばやと届いたアマチュアCGAコンテストの制作途中の画像と，DōGAで制作したX68000 XVIのイメージデモをお見せすることにしましょう。

前回テスト用の人体モデル（BOX君）が歩いている画像を掲載しましたが，そのデータを別のロボットに流用してみました。
この昔懐かしい思いのする悪役ロボット，結構気に入っています。

ジャブ！　ジャブ！　ストレート！
"明日のためのその1 打つべし，打つべし，打つべし"ということで，ボクシングのモーションデザインを行ってみました。

同様に，形状データを悪役ロボットに変更してみました。なかなか迫力あるファイトシーンですね。

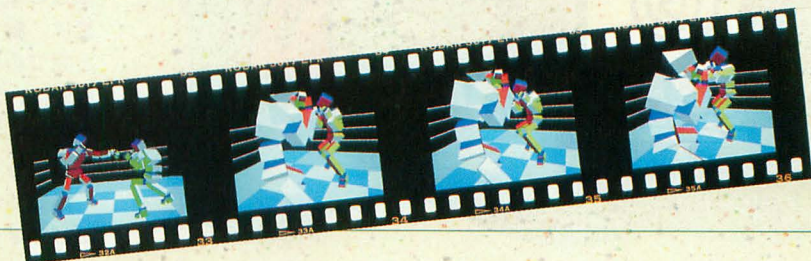
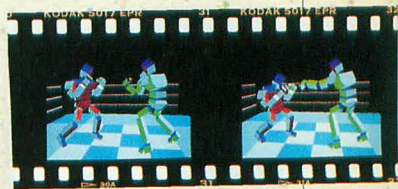
おまけに，前回裸で登場したGALに，ちゃんと服を着せて，ファイティングポーズをとってもらいました。
でも，彼女がところせましと暴れ回るのは，次回のお楽しみ。

次回のアマチュアCGAコンテストの募集も公表していないのに，すでに作品や制作途中の画像が送られてきています。これは，そのなかのひとつ，砂漠でのモビルスーツの死闘を描いた「ゴライアン（仮名）」の1カットです。すべてが黄土色の保護色でまとめられているあたりがシブい！

5月のシャープパソコンフォーラム'91に当チームから出展した，XVIのイメージデモです。おかげさまで，各方面からご好評をいただきました。すでに日本橋のJ&Pなどでその一部をデモしておりますが，近々再編集版を制作しますので，全国各地で見ることができるでしょう。

ボクシングのデータを利用して，いろんな角度からのカットを組み合わせるだけで，簡単に「ロッキー」の出来上がり。「エイドリアーン！」

[第4回]

止まる時間，流れる時間

寺尾 響子

「時間旅行はたぶん不可能だろう」……さきごろ京都で開かれた国際会議で，車いすの宇宙物理学者ホーキング博士が結論を下しました[1]。やっぱりタイムマシンはSF小説の中だけのものだったのかと思うと本当に残念です。宇宙にはワームホール（虫食い穴）が存在することが理論的には予想されています。ワームホールは宇宙の異なる時刻の異なる2点を結ぶ「抜け穴」で，この穴を人工的に広げて中を通れば瞬時に時間を移動できるというもの[2]。しかし，ホーキング博士によれば，ワームホールはとても不安定なので，すぐにつぶれてしまうのだそうです。

時間はゆっくりと，ときには速く，逆行せずに過去から未来へと川のように流れてゆくものなのでしょう。少なくとも現実の世界では。

けれど，ひとりになって自分のこころのなかを覗いてみると，過去も現在も未来も同時にあり，うずを巻いています。夏，浴衣を着て星空に花火の散るのを観ている5歳のころの自分がいました。年をとって体の自由がきかなくなり，天井を見つめながらCGに一生懸命だったことをおぼろげに回想している自分がいました。脳のなかのイメージの世界では時空を超えるのはとても簡単なことです。

響子 in CGわ〜るど

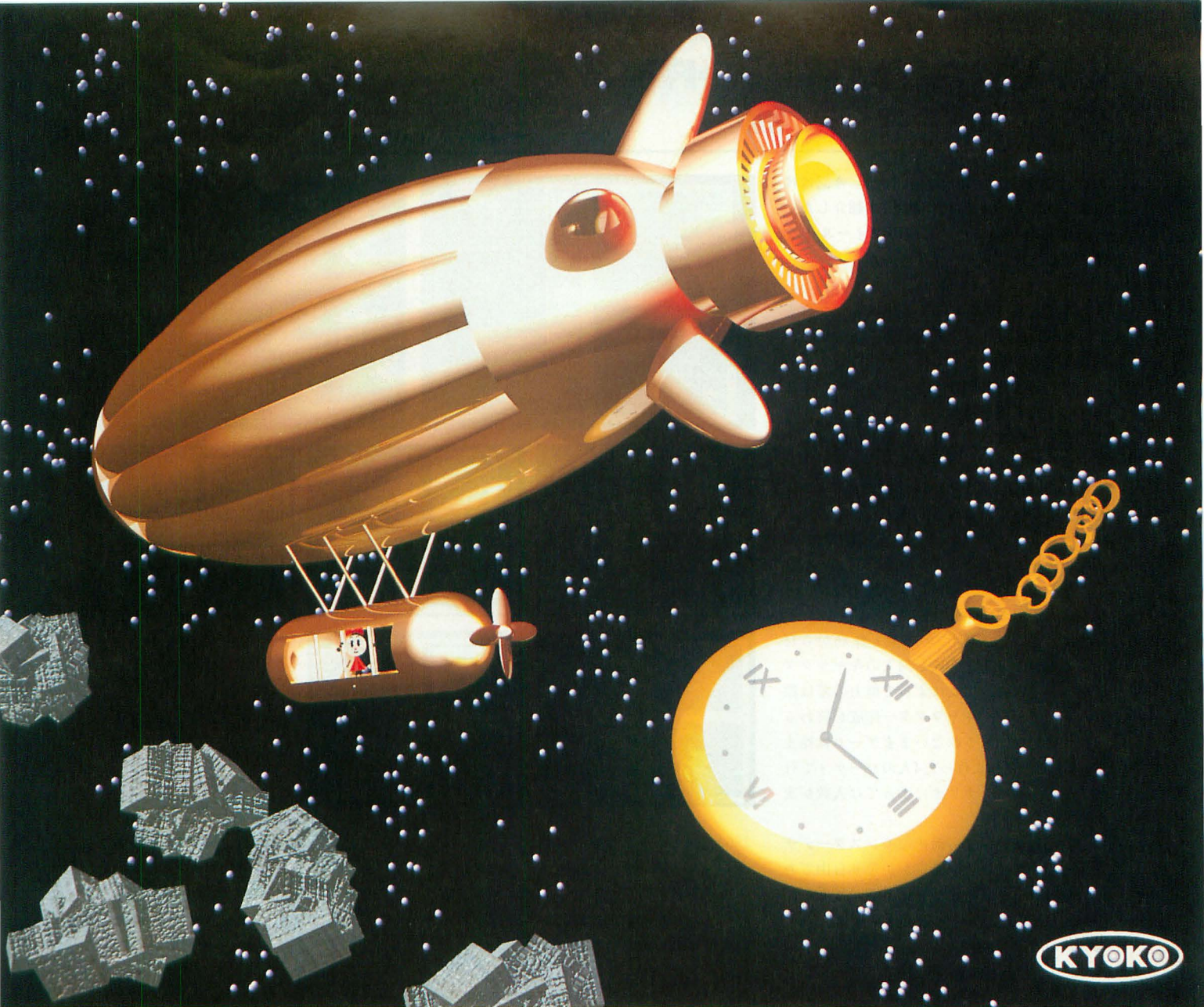
彼方から
時間が吹いてくる
風のように吹いてくる
目をつぶり　息をひそめて　待とう
顔に当たり　ゆっくりと　停止してゆくのが
わかる

静止画と動画

いざ，CGで作品をつくろう！　と思っても，はて？　何をどうしたらいいのかと迷う人が多いのではないでしょうか。静止画はきれいだけどなんだかものたりないような気もするし，かといっていきなり動画はたいへんだなあーと困ってしまいます。

いちばん大切なのは，「自分が何をどう表現したいのか」ということ。たとえば，花火を表現するとします。美しい色や形，花火と夜空のバランスを見せたいのなら静止画。花火が打ち上がって，きらきらと消えてゆく動きを見せたいのなら動画が向いているでしょう。じっくりと細かいところまで見せたいのなら静止画，物体は単純でも動きで見せたいのなら動画を選べばよいのです。

CGにおける静止画と動画の関係は，写真と映画のそれによく似ています。写真はひとつのテーマを1枚に集約して表現するもの[3]。一方，映画は動きのある流れでテーマを表現するものです。写真は，動かないので，はじからはじまでじっくりと見つめられることを前提として作られています。映画は，画像が次から次へと見るそばから消え去ってゆくので，記憶のなかに動きとして残るようにつくられています。

ふつうのアニメーションではセルに描かれた絵をぱらぱらめくって動きをつくり出します。3次

元CGのアニメーションも原理は同じこと。静止画像を大量にメモリに読み込んでCRT上で高速に切り換えるか，ビデオに静止画像をコマ撮りしたものを普通の速度で再生させるかして，絵を動かします[4]。

私はアニメーションが大好きです。動画をもっとつくりたいのですが，パワーがかなりいります。そこで，静止画でも前後の時間が想像できるような作品を心がけてきました。

時間旅行は不可能だろう，となるとタイムマシンを扱ったSFはつまらないものになってしまうのでしょうか。いままでに読んだSFのなかで，印象深いものの1冊にロバート・A・ハインラインの『夏への扉』があります。もはや古典ともいえる小説ですが，タイムマシンが重要な位置を占めています。でも，もういちど読んでも，よかったと思うに違いありません。なぜなら，テーマはタイム・パラドックスではなく，もっと大切なものだから。もし時間があったら，図書館で借りて読んでみてください。

---

1)　今年の6月に国立京都国際会館で開かれた「一般相対論に関する第6回マーセル・グロスマン会議」で，ホーキング博士は「タイムマシン」について否定的な見解を下しました。
2)　3年前に米カリフォルニア工科大のキップ・ソーン教授が「タイムマシンの可能性」について発表したもの。
3)　数枚で表現する組写真もあります。
4)　その場で計算させて動かすワイヤーフレームのアニメーションもあります。

久しぶりのTHE USER'S WORKS。今回は埼玉県のO/S softwareの作品を2本紹介したい。どちらもX68000シリーズ用のフィールドタイプ・ロールプレイングゲームだ。

●

なんでもX-BASICによる開発ということで，最初は速度などに懸念があったのだが，起動してみると心配するほどではないことがわかった。X-BASICでもコンパイルしてあればほとんど遜色はない。

両者とも特徴はシナリオにある。サブシナリオ重視で悪役の使い方などもいかにもそれっぽい。RPGに好ましいかは別にしても，流行路線に沿っているのは間違いないだろう。操作系はドラクエコンパチということだが，実際のゲームはファイナルファンタジーやファンタシースターっぽいように思える。

●Questland Stories

ファンタジー系の4つの職業と性別から自由にキャラクターを作成し，4人のキャラクターを作り上げる。男女によって能力の差は設けられていない。キャラクター作成が終わると特に目的は知らされないままゲームは始まる。基本的に最初に作った4人のパーティで行動するが，サブシナリオによっては人数が変動することもある。

512×512モードでの緻密なモンスターデザインはなかなかいい。設定がかなり自由である分だけシナリオの短かさに不満が残るが，全体的な完成度はかなり高いといえる。

●Ultimate Magic

Quest～に比べ画面は512×512モードから256×256モードに変更されている。ビジュアルシーンや画面全体の構成はいいのだが，フィールド移動時がさみしい。アニメーションパターンも改善の余地はある。前作がかなりの水準にあっただけに惜しまれる。

シナリオに関しては，まず「長くなって，目的がはっきりしたこと」が挙げられる。ストーリー性も上がっているので，全体の構成がわかりやすくなった。反面，容量の都合でオープニングデモが削除されたということで，マニュアルを読まなければ目的やバックグラウンドがはっきりしないことが残念だ。

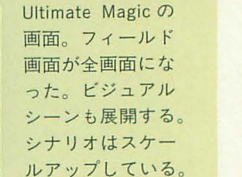
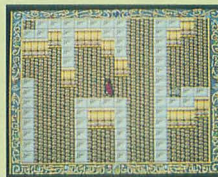
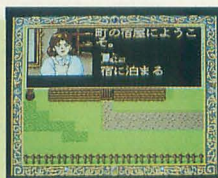
魔法の修得や武器の装備などにスキル制が取り入れられたことにより，「魔術師は剣が持てない」といったルールの理不尽さが解消されている。ルールは，基本的に金銭と経験をベースにしているので，かなり合理的な体系になっている。派手な魔法などでは味方を巻き込む場合もあり（モンスターも同様），戦闘方式がタクティカルコンバットになったことなどの公平なルール。キャラクターの体力で持ち物や装備に制限が加わるなど，よりリアル指向のゲームといえるだろう。

これらの入手を希望する方は封書に，

Questland Stories	1,600円
Ultimate Magic	2,200円

以上の額面（送料込み）の無記名郵便小為替を同封のうえ，下記まで連絡してほしい。

〒346 埼玉県久喜市2-2-7
大森方 O/S software

Questland Storiesのゲーム画面。キャラクターの名前は気にしないように。もちろん全部男のパーティでもかまわない。マップ画面が小さいがゲーム中はそれほど気にならない。

Ultimate Magicの画面。フィールド画面が全画面になった。ビジュアルシーンも展開する。シナリオはスケールアップしている。

注意：トラブル防止のため，タックシールまたは封筒に貼れるくらいの紙に返送先（自分の住所）を書いて同封してください。万一の場合に備えて別途に連絡先を記載したものを同封しておいたほうがよいでしょう。手紙は丁寧に，購入後は感想を送るのがマナーです。

SOFTWARE INFORMATION

残暑はゲームで乗り切れ，ということで今月もたくさんの新作が登場しています。発売されるゲームのジャンルも偏っていないので，自分の好きなものを選べます。また，記事には間に合いませんでしたが，「ドラッケン」もいよいよ完成のようです。

ボナンザブラザーズ

3分以内にお宝を頂戴するぜ！ との大胆不敵な予告をし，悪の町バッドタウンの夜を闊歩する正義の大泥棒「ボナンザブラザーズ」がX68000に登場である。オリジナルはセガの高精細グラフィックシステムのアーケードゲーム。今回の移植ではまるで盗んできたみたいに(笑)，グラフィックをはじめ，何からなにまでそっくりに仕上がっている。

ゲーム内容は簡単，悪の建物に侵入し，お宝を全部奪って屋上に脱出すればOK。しかし，それには悪の警備員や機動隊から隠れ，その攻撃をかわさなくてはいけない。もちろん相手を攻撃することもできるが，そこは正義の泥棒，気絶させるだけで敵を倒すことはできないのがミソだ。悪の銀行とか悪のデパートとかいう謎の設定が笑えるけれど，それにも増してキャラクターの動きもこれまた大爆笑間違いなしというところである。2P同時プレイもサポートされているので，文字どおりの兄弟泥棒コンビを結成することも可能だ。ぜひ正義のために盗んでくれ。 (八)

テカテカ頭にズン胴の体。天下御免のサングラス。こんなキャラクターが画面の中を暴れ回る「ボナンザブラザーズ」は，本当にきれいなグラフィックを実現している。その証拠に(？)，頭がテカテカ。ゲーム自体もとっても面白いし，なんといってもトラップが見事。バナナに空き缶，トランポリン。2人同時プレイでは基本的に協力するのだが，お宝は自然と取り合いになるので白熱すること請け合い。

X68000用 5″2HD版2枚組 価格未定
シャープ ☎03(3260)1161

なんとファランクスがあのパロに！

	(前回順位)
1.ファランクス	2↑
2.パロディウスだ！	1↓
3.イース	6↑
4.生中継68	4→
5.サイレントメビウス	7↑
6.遥かなるオーガスタ	3↓
7.黄金の羅針盤	ー再
8.A列車で行こうIII	5↓
9.ダッシュ野郎	ー初
10.ボナンザブラザーズ	ー初

永遠不朽の強さを誇るかと思われた「パロディウスだ！」ですが，なんと今月はトップの座から転げ落ちてしまいました。これで「パロディウスだ！」の連続首位は4カ月でストップということになります。

その「パロディウスだ！」を倒したのは，なんといおうか，やはりといおうか，ズームの「ファランクス」。登場直後は音楽やグラフィックの評判が高かったのが，最近では「動きがいい」「X68000でできることはすべてやっているような気がする」など，ゲームの本質的な部分に触れているハガキが増えています。それにしても，ゲームセンターと同じクオリティを誇る「パロディウスだ！」を倒してしまう「ファランクス」って，いったい……。

そのほか，「イース」や「サイレントメビウス」といった最近発売されたソフトが順調に評価を上げています。「イース」は電波新聞社のクオリティの高いアレンジがなされた移植を，「サイレントメビウス」は原作の忠実さを理由に挙げてくる人が多いようです。それと，「由貴ちゃんが好きだから」という輩がわいて出てきてますね。みんな好きだなあ。

9，10位は初登場でどちらもシャープのアーケード移植作。ダッシュ野郎には「意外に笑えるゲームだった」「夏に似合うから」(←ホントかい)という声が，ボナンザブラザーズには「完璧な移植で遊べそう」「家で遊ぶのには向いていると思う」などの声が届いてます。

さあ「パロディウスだ！」はこのまま落ちていってしまうのか？ オーガスタの挽回はあるのか？ 次回を待て。 (浦)

3D 2(仮称)

いまもっとも熱い注目を浴びているM.N.M.の新作。わけあって名前は出せないが，ワイヤーフレームによる「要塞惑星に十字翼戦闘機の決死隊が突撃して帝国を倒す」ゲームだ。オープニングには「とてもデカい戦艦」も出てくるぞ。どこかに似たゲームもあったがこれは完全な新作として制作されている。

とにかくワイヤーである。空中戦の滑らかさ，浮揚感，とにかく動きをお見せできないのが残念。だから，こう，クイっと曲がって，すぃーっと回って，シュイーンと，いうのが実に気持ちよく表現されているわけだ。

しかし，こいつの真価は動きだけではない。ゲーム性と見せ方が，いい。7,200円とお安め定価に超豪華スタッフ（ちょっと凄いぞ）。さらに，本当にこだわって作っていることがわかる演出の数々。そう，我々はこんなゲームを待っていたのだ。 (S.N.)

X 68000用 5″2HD版 予価 7,200円(税別)
M.N.M.Software ☎0423(60)3084

銀河英雄伝説IIDX+kit

ボードゲームの世界では既存のゲームの別シナリオだとか，上級ルールのセットとかいうのがわりと存在するようだけど，パソコンのゲームの場合はまだまだ少ない。せっかくゲームのシステムには慣れたのに，シナリオなどに飽きてしまってもうやらないというのは非常にもったいない。ボーステックから発売されている「銀河英雄伝説」はゲームの面白さに原作の人気が加わって，熱心なファンを得ているようだけど，このファンの支持に応え，「銀河英雄伝説II DX＋kit」が発売されることになった。グレードアップしたシステムで，新しいシナリオが10本楽しめるようになっている。システム的には各提督，各部隊の任意の惑星への配備，艦隊の初期構成とフォーメーションの変更，作戦中の艦隊の移動航路の表示といったことができるようになった。 (R.A.)

X68000用 5″2HD版 5,000円(税別)
ボーステック ☎03(3708)4711

飛翔鮫

全国1億1千万の縦スクロールシューティング(縦シュー)ファン待望の「飛翔鮫」が，ついにリリースされる運びとなった。複葉機を操り，敵の戦車や戦闘機をひたすら撃破していく熱いゲームである。オリジナルは東亜プランで，X68000版は金子製作所からの発売となっている。現在ゲームセンターを支配しているあらゆる縦シューの源となる地上空中同時攻撃や，ボンバーのシステムはすべてここから始まったのだから，この「飛翔鮫」の偉大さは，とても言葉では書き切れるものではない。必殺テクニック「切り返し」を駆使して自軍に勝利を導くのだ。金子製作所ではこの「飛翔鮫」を皮切りに「究極タイガー」「達人」「鮫！鮫！鮫！」などを続々と移植する予定なので，ファンの人たちにはたまらない強力ラインナップになりそうである。「飛翔鮫」が，君のX68000でよみがえる日は近い！ (八)

X68000用 5″2HD版 予価 8,800円(税別)
金子製作所 ☎03(5261)2147

シュヴァルツシルトII

「シュヴァルツシルトII」がやっとX68000にも登場。「シュヴァルツシルト」のときと同じく，他機種に比べて発売が後手後手に回っているので，いらついている人もいるかもしれない。しかし，あとから発売されるということは思考ルーチンやグラフィックがそれだけ改良されて発売されるということ。逆にとって，いいように解釈しよう。

今回はソマリ星系の新興星国であるオーラクルム国を操り，強大な国に育てていく。ゲーム開始直後の状況は，オーラクルムと友好関係にある小星国トリスティアが，好戦的なロッサリアに攻められているというところ。助けを出すか，それとも断るか？

戦闘シーンはもちろんのこと，さまざまなところが改良されている。戦艦も格戦闘タイプ，爆撃戦タイプと2タイプ（ともに8種類ずつ）登場する。また，小惑星の探索というのも重要なポイントになっている。 (R.A.)

X68000用 5″2HD版2枚組 予価 9,800円(税別)
工画堂スタジオ ☎03(3353)7724

ロードス島戦記 〜灰色の魔女〜

待ちに待ってた，やっと出る。そう，6月号の新作のコーナーでも紹介された，あのロードス島戦記がいよいよ目の前までやってきているのです。本格的なロールプレイングゲームがやりたかった人には本当にうれしいニュースですよね。現在の移植状況はというと，編集室に送っていただいたテストバージョンではイベントはかなりできていて，戦闘はもうチョイってところみたい。早ければ来月号ぐらいの号でレビューできるんじゃないかな（ねぇ，ハミングバードソフトさん，とプレッシャーをかけておこう)。見たところ，グラフィックは全部X68000用に書き直されてるみたいだし，システムの面でもディスクキャッシュが採用されたりと，ずいぶん頑張ってるみたいなので，期待大だね！ さあ，もうすぐX68000上をパーンが，スレインが，ディードリットが駆け抜けるぞ。小説版でも読んで待っててね。 (で)

X68000用 5″2HD版3枚組 9,800円(税別)
ハミングバードソフト ☎06(315)8255

ジーザスⅡ

ハレーすい星に乗ってやってきたエイリアンは，人間の脳からあらゆる情報を摂取し，環境に適応して形態を変えるというおそるべき生物だった。前作「ジーザス」で宇宙ステーションの訓練生，武麻速雄は死闘の末エイリアンをコンテナに閉じ込め，宇宙へと捨て去った。

Oh!Xの読者にはX1turbo版でおなじみの「ジーザス」。その続編がX68000に登場。「ジーザスⅡ」は，エイリアンの乗ったコンテナが地中海に落下，付近を航海中の貨物船がそれを引き上げてしまうところから始まる。その船に乗っていたのが銀河防衛軍予備校で速雄と同級生だった五色和哉。貨物船カリスト号で，また宇宙ステーション・ジーザスで，ふたたびあの悪夢が繰り返される。

200枚を超えるグラフィックと100種類のサウンドエフェクトで迫るストーリーはホラー映画を超える迫力だ。 (浦)

X68000用 5″2HD版 8,800円(税別)
エニックス ☎03(5272)2374

グループ・エックス

コムパックから3Dドライビング・シミュレータが発売された。このゲーム，プレイヤーが選べる車の多さがセールスポイントだ。NSX，テスタロッサ，ポルシェ928といったスポーツカーから，レックス，アルトワークスのような軽自動車まで，実に40種類もの車に乗れてしまう。それぞれのマシンはエンジン特性やギヤ比などがちゃんと本物どおりに設定されている。操作も本物の自動車の運転にそった複雑なものだ。

舞台は夜のサーキット。画面は3D表示で，街灯やセンターラインのあかりが迫ってくるぞ。ゲームにはタイムトライアルの予選と5台の車で争う本選がある。本選では，ゴールした順位によってポイントが与えられ，獲得したポイントとベスト・ラップはセーブすることもできる。X68000版はBGMもセールスポイント。内蔵FM音源のほかにMT-32/CM-64/32L，SY22/TG33，そして，SC-55が使え，それぞれ別の音楽が用意されているぞ。 (浦)

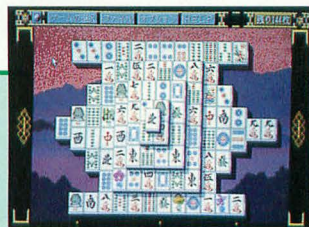
X68000用 5″2HD版2枚組 7,800円(税別)
コムパック ☎03(3370)3401

スーパー上海ドラゴンズアイ

パズルゲームブームに先鞭をつけたといわれる，アメリカ生まれの大ヒットパズルゲーム「上海」の続編が誕生した。アメリカのアクティビジョン社から発表された，「スーパー上海ドラゴンズアイ」がそれだ。従来の上海をパワーアップし，さらに新しいルールのゲームを加えている。従来の上海のモードでは牌が麻雀牌のほかに7種類のデザインの牌が加わって画面にバリエーションを加えている。また，配置も従来の形に加えて，十二支から取った12種類の配列が登場。さらに自分で好きな配置を作れるコンストラクションモードもついたのだ。

新しいゲーム「ドラゴンズアイ」は2人でプレイする対戦型ゲームだ（ひとりのときはコンピュータが相手)。ルールはおおまかにいって，牌を崩す「スレイヤー」と牌を置いていく「マスター」の対決。なかなか奥が深いものになっているぞ。

X68000版の登場は少し先になりそうという話だが，PC-9801ではMIDIにも対応していることだし，きっとそれ以上に頑張った作品になることだろう。 (浦)

X68000用 5″2HD版 7,800円(税込)
ブラザー工業(TAKERU) ☎052(824)2493

画面写真はPC-9801用です

ついに甦る冒険と感動

Saitou Susumu
斎藤　晋

2年以上の歳月をかけ，ついに完成したX68000版イース。それもイース1だ。超高密度のグラフィック画面と大胆なアレンジでイースの世界を演出する感動の意欲作。さあ，新たなイースを語るのは君だ。

時は流れ，イースのことを語り継ぐ人も少なくなってきた。美しい2人の女神のこと，クレリアという金属に秘められた力のこと，アドルという赤髪の少年のこと。

そう，多くの人の思いがそこに集まった。そして，与えられた数々の賞賛の言葉。イースは……，う，うう，誰だぁ！「イースはいーすよ」なんてつまらん洒落をいった奴は。イースは私の青春だったんだぞ。それから，この際だからいっておくけどイースというのは，イース2で完結していて，ワンダラーズ・フロム・イースは別のソフトだからな。その点はくれぐれもよろしくな。いいな。頼んだぞ。まったくぅ。

これが電波のイースだ ◆◆◆◆◆◆◆◆

というわけで，X68000のイースは発売された。店頭に山と積まれたパッケージを前に私はしばし立ちつくしていた。電波新聞社からイースが出ると聞いて以来，どれだけ待ったことか。間違いない，たしかに電波のイースである。箱だって正方形だ。

ところが，このイースはこれまでの電波新聞社の移植作品とは一線を画すものだ。これは決して忠実な移植ではない。なんとアドルが八頭身，なわけはない（それは噂にすぎなかった。ただし，三等身半ぐらいはあるようだ）。

とにかく，イースをよく知っている人はもちろん，初めての人でも試してみればきっと仰天するに違いない。これはイースの世界をX68000の機能を十分に生かしたクオリティの高い表現，演出でリメイクしたものなのだ。そしてX68000の表現力をここまで引き出した作品は初めてだろう。

驚くべきはグラフィックだ。画面は512×512ドットの256色モードで，町の外壁，地面，建物の細部，そして樹木にいたるまでが執拗なまでに丁寧に描かれている。すごい。いや，凄い！　石畳はごく自然に汚れているのがわかるといったリアルさだ。しかもスクロールは異常に滑らかだ。

そして水面に注目，最近は常套手段となりつつあるラスタースクロールが使われているようだが，その使い方がなんとも粋なのだ。決して，これ見よがしなウネウネとした波ではなく，水面の細かい波が静かに揺らぐさまが見事に表現されている。

でもって，登場キャラのグラフィックが実にエグイ。占い師のサラは美しいし，盗賊のゴーバンさんはおっかないぞ（なんといっても，この顔にイース2での台詞をオーバーラップさせると結構おかしい）。で，肝心のフィーナだけど，……（いいんだ。私にはまだレアさんがいるんだし）。ちなみにどうでもいいけど武器屋のおやじはどうみたって「リチャード・キンブル，職業医師」だ（逃亡者って知ってる？）。

また，シナリオやマップも見直しがなされた。オリジナルを知っている人はそれぞれに思い入れがあるだろうから異論もあるかもしれない。が，少なくとも物語を浮き上がらせることに成功している。余計な経験値かせぎによる冗長さはほとんどなくなったようだ。

といったところで，物語を紹介しよう。

伝説は語る ◆◆◆◆◆◆◆◆◆◆◆◆◆◆

かつて，この地に栄華を誇った理想郷，それがイースだ。イースは2人の女神と6人の神官によって治められ，クレリアという金属によって繁栄をきわめた。しかし，突如襲ったおそろしい災難によってすべては無に帰した。人々は国と共にこの地を離れ，2人の女神もその姿を消したという。

これが，ゲームの背景となるイースの伝説であり，その歴史を記したものがイースの本である。イースの本は全部で6冊あり，それぞれの章には6人の神官の名がつけられている。

物語は，主人公の少年アドルがエステリアの港町ミネアに着いたところから始まる。町の背後にはカルデラ式の火山があり，麓からは銀が掘り出されたが，なんらかの理由で魔の封印が解け，怪物たちが徘徊するようになった。鉱山は封鎖され，町はすでに沈黙していた。そして，一連の出来事の背後には魔力による支配を目論むひとりの男の影があった。占い師サラの言葉により，

X68000用　5"2HD版2枚組　9,600円(税別)
電波新聞社　☎03(3445)8201

新たにデザインされたミネアの町

見張り台にいるのは詩人のレア

アドルは自分がこのおそろしい野望と対決する運命にあることを知る。また，そのためにはイースの本を探し出し集める必要があった。

さあ起動だ。天野喜孝のタイトルグラフィックが名曲“FEENA”とともに，おごそかに画面に映し出される。フィーナである。ちなみに私は天野さんのファンでもなんでもないが，この絵は美しい。パッケージの絵を見ると髪の毛の細かい線が気に入らないけど，画面で見ると綺麗にぼけて本当に美しい。ああ，だめ。もう生身の女性を愛せなくなってしまいそう。が，見惚れているといつまでたってもゲームは始められない。未練を残しつつスペースキーをたたくと，アドルはミネアの町の入口に立っている。

街並みは，オリジナルとかなり様子が変わっている。シナリオもディテールに若干の違いがあるので，他機種版でやったことのある人でも，街の人との会話をはしょらずにチェックしておくこと。イースはアクションRPGの姿を借りたアドベンチャーゲームのようなものだから物語を味わうようにプレイしたい（だからぁ，ワンダラーズ～は違うんだって）。

アドルの戦い ◆◆◆◆◆◆◆◆◆◆◆◆◆

町の外の草原で若干の経験値とゴールドを手に入れたら足りない装備を整えて再びサラの家を尋ねる。イースの本がサルモンの神殿の地下にあることをクリスタルが示しているという。いよいよ本格的な冒険の始まりだ。

リアルな滝。ラスタースクロールの妙技だ

水辺に立つサルモンの神殿

フィーナが振り向く

お馴染みの登場人物。サラ，ジェバ，フィーナ，ゴーバン

神殿に入ると最初のボスが登場

6番目のボス。メタボールか？

サルモンの神殿はカルデラ式火山の湖の縁に建っている。神殿への道は完全に描き直されており，岩場はフラクタルかと思うほど自然だ。ともかく神殿の前まで来たらセーブして衛兵を突破しよう（素早いから要注意)。扉が開き，中に入るといきなり最初のボスがお出ましになる。

結局，オリジナルからアプローチのダンジョンが省略されたことになるが，この部分の音楽はあとの地下牢のある迷路に振り分けられている。アップテンポな戦闘曲に変わるのはさらに地下に降りたときだ。曲だけでなく敵の動きもダントツに速くなる。迷路自体はオリジナルといっしょだが，物語の流れをうまく盛り上げる刺激の効いた演出だ。

無事にフィーナを解放し，あるアイテムで見えない通路を見破れば神殿の最下層へと進めるだろう。腰まで水につかりながらも奥へ進むと，予想どおり第2のボスが待っているというわけだ。

これで君も大丈夫 ◆◆◆◆◆◆◆◆◆◆

このあと，アドルは廃坑，そして魔力で作られたダームの塔へと進んでいく。深みのある廃坑のグラフィック，装備した武器によってアドルのグラフィックが変わるのも楽しい。

イースには難しい謎はほとんどないが，1カ所だけ必要なアイテムを取り忘れるところがあり，それはマニュアルにも記されている。イースの本が読めるようになったら必ず注意深く読むこと。そしてダームの塔では，出会う人の名前と居場所を忘れないことだ。

エンディングは派手なアニメーションこそないが，静かで味わいがある。そしてタイトルロールは文字のみ。だが，この文字がただものではない。開発スタッフの技術と思い入れがひしひしと伝わる一作だ。

さあ，オリジナルのイースを知っている人も，初めての人もX68000ユーザーならこの作品を買ってプレイしよう。そして一緒にイースについて語ろうではないか。

イースは永遠に不滅です

イースはパソコンゲームの古典として歴史に残る名作だろう。イースは1987年に発表された日本ファルコムの作品だ。美しい物語と優しいゲーム性，そして素晴らしい音楽で多くのイースファンを生んだ。

基本がしっかりしているから，十分に吟味されたうえでの演出の違い，解釈の違いは，新たなイースの魅力を引き出すことができるはずだ。電波新聞社のイースはそのことを証明した一作だ。オリジナルとは異なる部分も多いが，移植というより，まったくのリメイクとしてプレイする価値がある。私は将来，3Dのリアルタイムアクションでイースをやってみたい。

え？ イース2はどうなるのかって？ それは私も気になるなあ。オープニングのリリアはどんな顔で振り向くのだろう。

総合評価	0　5　10
グラフィック	★★★★★★★★★★
音楽	★★★★★★★
演出	★★★★★★★★★
技術	★★★★★★★★★★
感動	★★★★★★★★

THE SOFTOUCH ●生中継68

プロ野球,欠かせないのは中井美穂

Ogikubo Kei
荻窪 圭

「パワーリーグ」や「ワースタ」など，X68000にもいくつか野球ゲームはあったけど，他機種でも発売されていたし，決定版とはいえなかった。しかし，この「生中継68」は，まさにX68000での野球の決定版という出来だ。

まずは気の早いところで，

“祝！ 中日優勝！”

ときたもんだ。ここまできたら，もう巨人は完全に脱落だから，あとはヤクルトを抑え込めばいいのである。うーん。いまはオールスター前だけど，優勝宣言しちゃうのである（本誌発売時に首位から脱落していたら笑ってくれ）。

オープン戦で慣れよう◆◆◆◆◆◆◆◆

とりあえずは，練習の意味でオープン戦である。ここでカンを摑んでから，ペナントレースへ突入するのだ。ペナントレースは時間がかかるので，紹介は次号。

はやる気持ちを抑えつつ，中日（っていっちゃいけないのか。名古屋ドンジャラスだな）を選択する。相手はとりあえず，巨人，じゃなかった東京ガンバルズである。

オープン戦だから，オーダーも適当でいいや。それどころか，オープン戦とはいえ，相手チームのオーダーまで変えてしまうことが許されているのだ。ゲームだもんな。というわけで，プレイボールだ。

●打つこと

男の3種の神器，“飲む・打つ・買う”に含まれるくらいだから，打つことは重要である。うぐいす嬢にアナウンスされて，たつなみが左バッターボックスに入る。写真のようにストライクゾーンとヒッティングスコープが出る。これがポイントだな。主審の視点だ。

投手がボールを投げると，ボールが来るポイントが野村スコープのように表示される。ボールが来たら，ジョイスティックでヒッティングスコープを動かし，ボタンを押して“打つべし”である。生中継68のすごいところは，この，ヒッティングスコープだ。うまいやつは，多少ポイントからずれていても打てるし，下手なやつはちょっとずれただけで，ポップフライを打ち上げたり，ぼてぼての内野ゴロになったりする。ヒッティングスコープは長距離打者なら縦長だし，広角打法の安打製造機は横長だ。大きさもバッターのテクによって異なるから，打っているほうも楽しい。

●投げること

先発は西本だ。投げるときは，センター方向からの望遠レンズになる。真後ろだと当然捕手も打者も見えないので，右投手の場合はちょっと左翼寄り，左投手の場合はちょっと右翼寄りになる。当然であるが，走者がいないときは降りかぶるが，いるときはセットポジションである。

球種は，ジョイスティックの方向で決める。得意不得意いろいろあるが，変化球が曲がらないやつはほとんど曲がらない。曲がるやつはグインと曲がる。西本のシュートなんて，しゃれにならないくらい曲がる。球種を決めたら，Aボタンで投球動作に入る。ここから，ボールをリリースするまでの間にジョイスティックでコースを決める。投球動作に入ってからコースを決めるという技を使うのだ。さすが，プロだな。

投げ方もいろいろとバリエーションがあり，投げているほうも楽しい。野茂だけ専用のフォームがあるのは，贅沢である。

●守ること

投げて打ったら守らねばならない。打者が球を打つと，打球が飛ぶわけだが，打球ってのは速いから，あっという間に画面から消える。すると，打球の強さ，方向によって画面が切り替わる。内野か外野か，中堅方向か左翼方向か右翼方向かで，2×3＝6種類あるのだ。打球の飛び具合からどの画面に切り替わるかを予測し，身構え，とっさに反応するのがプロである。難しいのは，ジョイスティックを倒した方向に野手が動くということだ。中堅方向から見るときはそれでもいいが，左翼や右翼はジョイスティックを前に倒したからといって，野手が前に動くとはかぎらないのだ。

で，まあ，野手が捕球するわけである。守備がうまければ多少正面からずれていても捕球してくれる。打球が少し遠ければ，Aボタンでダイビングキャッチすればいい。けっこうなんとかなるものだ。

捕球したら送球する。こいつがまた難しい。というのは，左翼方向から見ているときは左下，中堅方向だと左，右翼方向だと左上が一塁になるのだ。間違えて本塁に投げてしまったりすると悔しい。が，何にもしないと，一塁（一塁にランナーがいると

X68000用 5"2HD版2枚組 9,800円(税別)
コナミ エンタテイメント ☎03(3264)5678

ヒッティングスコープがわかるかな。2人用の画面

二塁）を指しているので，下手に動かさないほうがいい結果が生まれたりする。

フライは落下地点にマークが出るから，マークのちょっと前で待つ。マークの位置は落下地点だから，マークの上で待っていても足元に球が落ちてヒットになるだけだ。ちょっと前というのがポイント。

●走ること

世にあるあまたの野球ゲームで，いちばん洗練されていないのが走塁技術だ。どのゲームでも「こいつらプロか？」と思うくらい走塁が下手である。生中継68でも同様だが，A＋Bボタンで走者が塁間で止まる，という技が追加されたのは評価できる。

が，とにかく走塁は難しい。もう少しインテリジェントに選手が動けば，“外野フライでゲッツー”という情けないプレイもなくなるだろう。塁間にはさまれたら100％逃げ切れないと思ったほうがいい。

●ホームランの打ち方

長打力のあるバッターの場合，バットの芯の少し上で打つ。すると，ボールは上に上がるから，あとはタイミングでホームランだ。ボールをうまくバットに乗せるのだな。これがまた難しいのだが，狙って打ったホームランの味はまた格別である。

●試合終了

試合が終わると，プロ野球“の”ニュースの放映である。そのオープニングがまた秀逸なので，大笑いである。んでもって，ミポリン似のねえちゃんとユッキー西本の解説を聞いて，1試合が終わる。

ピッチャーくわわった，投げました

ボールの落下地点は照準のように見える

チームエディット画面

今日の取り組み，じゃなくて対戦(ペナントモード)

ときめきの新チーム◆◆◆◆◆◆◆◆◆◆

さて，気を取り直してエディットである。好きなチームを作る。フランチャイズを決め(地図から都市を選ぶのだ)，チームカラー，ユニフォームを選んで，球団マークを選んで，チーム名をつける。これで球団の出来上がりだ。凝っていること凝っていること。

生中継68がX68000専用だなあ，と思わせてくれるのは，チームエディットモードでは“マウスを使って操作する”ことである。ちょっと使い勝手はいまいちだが，ジョイスティックやキーボードに比べると，抜群の操作性だ。

チームが決まれば，選手のエディット。名前を決めるモード，力を配分するモード，ボーナスポイントを配分するモードがある。なかなか難しいのは，力の配分だ。バランスよく配分しないと，勝てるチームは作れない。作ったチームはセリーグかパリーグに組み込んで，一緒にペナントレースを戦うことができる。

ファールボールにご注意ください◆◆◆

BGMはちょっとうるさい。応援のトランペットがうるさいのだ。昔から応援のトランペットはうるさいと決まっていたので，非常に忠実だ。こいつもPIXY99Xのスタジアムモードで聴くと，思わずビールに枝豆が欲しくなるほどの臨場感である。歓声にかき消されて，うぐいす嬢が「ファールボールにご注意ください」と叫んでいる。

そろそろ終わりだから追加しておくが，対戦もできる。対戦モードでは，左と右に画面が分割されるから，それぞれ自分の必要なほうを見て戦えばいい。人間同士だと微妙な駆け引きが出てきて面白いぞ。

さすがにさすが◆◆◆◆◆◆◆◆◆◆◆◆

コナミはさすがである。ちょっと処理が重かったのか，動きがばたばたしている感はあるし，リアルな画面にするためか選手が小さくてよくわからない，というのはあるが，あれだけ見せるゲームを設計するのはさすがだ。演出はいうことがない。細かい欠点はいろいろあるが，スポーツゲームだけは凝りだしたらきりがないから，生中継69，じゃなかった，生中継68 '92に期待しよう。打撃10傑が見られるとか，プロ野球“の”ニュースで「今日のホームラン」が見られるとか，走塁をもう少しインテリジェントにして，バッターがバントの構えをしたら自動的にスタートを切るとか，フライを上げたら，塁間で様子を見るとかしてもらえると，うれしい。

プロ野球が好きなら，買うべきである。ワースタとも，パワーリーグとも，やじうまペナントレースとも，どれとも異なる新しい魅力の野球ゲームだ。でも，いまいちばん面白い野球漫画はモーニングの「レジー」だ。

見せているところを見なさい

生中継68っていっても，'68年の野球をシミュレートするゲームじゃないぞ。データは91年開幕時のものだ。当たり前だ。さて，X68000ならでは，というスポーツゲームがなかなかなかったから，それだけでもこいつはうれしい。X68000用のアクションゲームは動きの滑らかさとか，スピード感を追求するものが多かったが，この生中継68は違う。動きは滑らかとはいえないし，操作もやりにくい。しかし，である。プロ野球ファンなら誰でもつい目が真剣になってしまうだけの“愛”があるのだ。細かいことにうだうだいうやつは，心が狭いのだ。自分で遊ぶのに飽きたら，観戦モードにしてBGVにするのもいいぞ。生中継68はA級大作ではないが，私は支持するのである。はっはっは。中日が強いと調子がいいわい(談)。

総合評価 (0 — 5 — 10)

項目	評価
プロ野球度	★★★★★★★★
動きの滑らか度	★★★★★
応援のうるささ度	★★★★★★★★★
アーケー度	★★★★★★★★
野茂のフォーク	★★★★★★★★★★

封印はお手やわらかに

Yaegaki Nachi
八重垣 那智

「アークス」シリーズ番外編とでもいうべきこの作品。番外編というと「あーくしゅ」を思い出しそうだが，このゲームは真剣勝負のシューティングゲームとなっている。ストーリーはあまり気にしなくていいみたい。

私は「アークス」に対しては，いい思い出がない。たしか，PC-8801でプレイを試みて，わけもわからずに挫折した覚えがある。そんなトラウマもあり，「アークス2」や「あーくしゅ」などには一切手を出さなかった過去を持っている。勇気を出して嫌いなものを食べたら，おいしくなくてもっと嫌いになってしまい，一生食べなくなってしまうというような感じに少し似ていると思っていいだろう。

ところが，この「アークス・オデッセイ」はストーリー的なつながりは特に関係なく，とにかく「シューティングゲーム」であるという言葉につい誘われてしまい，気がつくとジョイスティックを握っていたのだ。実に不思議なものである。

アークスじゃないアークス◆◆◆◆◆◆

このゲームは例の「クォータービュ～」という「アクシス」でお馴染みの視点を採用している。その「アクシス」に馴染めなかった私は最初は不安になったが，これ自体は逆に立体感の表現に適していて，プレイしていくうちに，しだいに馴れてくるので安心していいだろう。感覚的には，「メルヘンメイズ」に近いという気がする。

ゲーム内容はいたって簡単。フィールドでのイベントをクリアして，どんどん扉とかをくぐり抜けていけばボスが登場，倒せばその面はクリアである。8面抜ければ結末までたどりつけるのだが，あくまでイベントといっても「鍵を探す」といったようなアイテム探索がメインだから，床を1ドット単位で調べたりする必要は一切ない。そういった点から考えると，これは純粋なアクションゲームである。それでいてタイミングを必要とする操作を要求されるような場所もないので，非常に快適な展開が見られる。こういったノリのよさはこのゲームのいちばんいいところであるといえるだろう。

ただ，体力を失ってゲームオーバーになった場合は，面の最初から完全にやり直しになる。長い面のボスなどで力尽きた場合は繰り返しが苦痛になることもあるかと思われ，そういった点の配慮が乏しいのは少し残念なことである。とはいえ，継続の回数に制限はないのだから，繰り返して覚えれば抜けられるというように考えて，納得するしかないだろう。また，途中でプレイをやめる場合にはパスワードがあるので，あとで何度も挑戦することも可能である。別々に育てたキャラクターを持ち寄って同時プレイすることも，パスワードを利用して実現できるので，その点はよく考えられているといっていいだろう。

操作は8方向移動にボタン2個の定番型で，ショット（防御）とアイテム（魔法）といった見なくてもわかる方式は評価できるのだが，アイテムなどのセレクトがボタン両方でウィンドウを開く方式なので，やや不便である。これに関してはキーボードを用意して，ESCキーを使ったほうが確実なので，じゃまだとは思うがキーボードを手近なところに置いておくことをお勧めする。まさに生活の知恵である。

サウンドはもちろんウルフチームだから，お約束のMIDIにも対応している。ただ，いまどき起動時にキーを押させて手動で判定するのは，ちょっとなさけないので工夫してほしいところではある。いちおう効果音でプレイヤーの悲鳴が出るのだが，馴れないと少々気持ち悪いかもしれない。できれば，敵のボスとかも高笑いをしたりしてほしかったが，そこまでは高望みのようである。

X68000用　5"2HD版2枚組　8,800円（税別）
ウルフチーム　☎03(5394)5565

戦士の過去と事情◆◆◆◆◆◆◆◆◆◆

プレイヤーは最高2人まで同時プレイでき，キャラクターは4人の中から選んでプレイする仕組みになっている。4人は武器も違えばその戦う理由も異なっていて，それぞれがそれぞれのドラマを背負っている。これらは説明書やゲームスタートでもわかるし，直接ゲーム中には影響しないので，選択するときはキャラクターのゲーム中の特徴を考えたほうがいい。セレクト画面で左から順に説明していこう。

★ジェダ・チャフ

三日月型のショットを打つ。若干射程に制限があるが，威力や連射数はバランスが取れていていい。魔法は防御の魔法で耐久力を上げたり，炎上して多量なダメージを与える強力な呪文が使える。

★エリン・ガーシュナー

チェーンソードというムチのような鎖を振り回す。威力はかなり強いが，振り回すために連続でダメージを与えにくいのが欠点。その代わりに壁の向こう側も攻撃可能。魔法は弱いが体力に優れる。

★ディアナ・フィレリア

矢を連射するがいまひとつ威力に欠ける。矢は壁で反射するので，死角から敵を攻撃

牢屋には当然カギが必要だ

できるのが利点。レベルを上げれば3方向に射てるようになり，狭い場所では有利。体力回復の魔法を使えるが，攻撃魔法は持続性が乏しく不利。

★ヴィド・シア

魔法は強力なのだが，あまり使い勝手はよくない。ただし，魔法のアイテムの威力もほかのキャラクターより強力になっている。通常時のショットの連射力が弱くて扱いにくく，体力も最も貧弱なので厳しいと思われる。

といった面々だが，私のお勧めは騎士のジェダである。これでプレイするのが，最も楽だと断言してもいいくらいだ。もっと楽をしたいのなら連射つきのジョイスティックを用意するといいだろう。このゲームはシューティングゲームということを忘れてはいけないのである。

YとNの落とし穴◆◆◆◆◆◆◆◆◆◆◆

シューティングゲームとはいえ，このゲームではメッセージも大事にしなくてはいけない。イベントに際して会話が重視される傾向があるため，できるかぎり人の話を聞くことが肝心である。場面によってはフラグになっているので，気がつかないとハマってしまう可能性もあり，油断がならない。また，歩きながらショットを射ち続けていると一瞬でメッセージウィンドウを消してしまうことが多いので，常に注意していなくてはいけない。

さらに，その会話の中でプレイヤーの返事を求められることがある。これはよく考えてちゃんと答えないと，最悪の場合は一瞬でゲームオーバーにされるものがある。これはまったくもって凶悪な罠としかいいようがない。

運命のはい，いいえ。信用してはダメ

なぜか味方をする２匹のモンスター。謎だ

アイテムを拾うか否か程度ならばやり直しがきくからまだいいのだが，ほかの場合はちょっとやっかいなことになるので注意が必要だ。通常そのままボタンを押すとデフォルトで「はい」を答えてしまうので，「いいえ」を答える場所では連射ボタンを使わないなどの工夫をすることが望ましいだろう。一度読んだメッセージを飛ばしているときによくやってしまうのだが，実際にゲームオーバーになってみたりすると，このうえなく不快である。この点に関しては許しがたいものがあるといえよう。

しかし，この会話メッセージのおかげでゲーム自体の流れがわかりやすく，イベントの把握も容易になっているわけだからあまり非難することは酷であるのかもしれない。あっちを立てればこっちが立たずといった事情があるのだろうが，もう少し工夫してほしいところではある。あと注意することは変な語感の固有名詞を声に出して読むと笑ってしまい，せっかくの気分が台なしになるので絶対やってはいけないということである。これは本当。

全体マップ。上の惑星まで？

キャラクターはどれもアークスでお馴染み

戦いの果て◆◆◆◆◆◆◆◆◆◆◆◆◆

ゲーム中に気になった点をいくつか挙げておく。味方を連れていない場合のシナリオ処理が一部おそまつといったところや，火の面での異常なスローモーションはもう少しきちんとしてほしかったと思う。特に後者はあまりに露骨なので，シナリオを考えてうまく処理するべき問題である。最初は急に遅くなったことに思わず動揺してしまったので，心臓によくないような気がして心配である。

さて，実際にプレイするとわかるが，ジェダをプレイヤーに選んでいれば，このゲームはさほど難しいゲームではない。強い敵にはレベルの高い魔法を効果的に使い，アイテムをしっかり集めれば，エンディングにはすぐにたどりつけるだろう。一長一短のキャラクターたちというよりも，キャラクターの違いが難易度の違いになっているので，やさしいのに飽きたならば別のキャラクターを選んでプレイすればいいのである。

ほとんど試せなかったが，2人同時プレイで接待用に使うのも一興であると思われる。手軽な分だけ繰り返し遊ぶゲームと考えるのがいいだろう。そういった点では「遊べる」ゲームであると見ていい。いろいろなパスワードを集めて解析したりすることもやってみると，強力なキャラクターや変なキャラクターができたりして，案外面白いかもしれない。

とにかく，こういった感じのお手軽なゲームはあまりなかったので，私としてはそのあたりを評価したい。ゲームだけにかぎらず，繰り返し楽しめるものはいいものなのだから。

アーケードライクで二重丸

プレイ中に常に考えていたのは，アーケードゲームの「ダークシール」の影響がやたらと感じられることである。キャラクターセレクト画面とか鎖を振り回すところなど，思わずびっくりしてしまうくらい似ている。たぶん私と同じように，あのゲームが好きな人がスタッフに絡んでいるに違いないだろう。まあ，いいところだけをうまく取り入れているので，文句をつける気はないが，もっと大胆に演出面なんかも取り入れてほしいような気もする。似ているといえば，2面は「魔獣の王国」の9面に似ているのだが，こう書いてもわかる人があまりいないだろうから，これは蛇足だったか。うんうん。

総合評価	0　5　10
難しさ	★★★★★
ゲーム性	★★★★★★★★
技術力	★★★★★★★
アークス度	★★★★★
ビジュアル	★★★★★★
中毒度	★★★★★★★★★

お茶を飲んで天下を取ろう

Urakawa Hiroyuki
浦川 博之

「信長の野望」シリーズにまたまた新作が登場。その名も「信長の野望・武将風雲録」。今回は茶器，鉄砲，鉄甲船などがメインとなっていて，それを持っているか否かが勝負のポイントになる。お茶会を開けば，先が開ける？

おしもおされぬ超有名戦国シミュレーションゲーム「信長の野望」シリーズに最新版，「武将風雲録」が登場した。前作「戦国群雄伝」から約2年ぶりの登場ということになる。

「あれ，もうそんなになるんだっけ？」と感じる人も多いと思う。光栄にはもう1本の歴史シミュレーション「三国志」シリーズがあり，互いに影響を与えながらバージョンアップを繰り返してきた経緯がある。去年は「三国志II」がリリースされていたために，「信長の野望」シリーズのほうにはあまり世間の注目は向かなかったというわけだ。まあそれにRPGやアドベンチャーと違って，すぐに次回作がほしくなるというものでもないので，そんなにひさしぶりという気がしないのかもしれない。

わざわざニューバージョンを出すというからには，それなりの新趣向が盛り込まれているに違いない。まずはそのあたりからチェックしてみよう。

政治家は文化人たれ◆◆◆◆◆◆◆◆◆

光栄が唱えるこのゲームのキーワードは「文化と技術」だ。戦国時代には茶の湯が千利休によって完成されているし，鉄砲の製法が伝えられて天下統一に大きな役割を果たしている。これらを戦国時代を語るうえで欠かせないと考え，大きく取り上げたのが「武将風雲録」というわけだ。

まず，技術。国ごとに技術力というパラメータができた。技術があるとどうなるかというと，金山の採掘ができるようになる。採った金は軍資金として使えるのだ。そして技術が高くなるにつれて鉄砲の製造，鉄甲船の建造ができるようになる。この2つはゲーム中でハイテク兵器として絶大な威力を誇る。海上でしか使わない鉄甲船はともかく，鉄砲はゲームを有利に進めるために欠かせない。堺の商人から買うこともできなくはないが，商人と大名との間には友好度が設定されており，仲が悪いと売ってくれないこともある。コストの面からいっても技術力を上げて，自前で製造したほうが得だ。

そして文化。これもまた国ごとに文化度というパラメータがある。技術革新の速さは，この文化度によって違ってくるようになっている。大名だけが技術の大切さに気づいていても，国が田舎では思うにまかせないわけだ。

では，文化度を高めるためにはどうしたらいいのだろうか。ここで茶の湯が登場してくる。当時のインテリジェントなイベントである茶会を開けばいいのだ。茶会を開くには当然茶器が必要で，これまた堺の商人，今井宗久から手に入れる。ところがこれも仲が悪い大名には売ってくれない。それだけ茶会を開くまでの道は険しい（はじめから持っていれば話は別）わけだが，その分メリットも大きい。茶会を開くと参加した武将の教養が上がる。教養がある武将は外交や取引を有利に進めることができ，天下統一に大きく貢献してくれるのだ。

いままでの「信長の野望」では，とかく国力さえつけてしまえばOKという感じだったが，この「武将風雲録」ではこういった技術と文化の要素を加えることによって単調になりがちなプレイにバリエーションを与えている。

X68000用 5"2HD版3枚組 9,800円(税別)
光栄 ☎045(561)6861

そのほかの主な変更点としては，舞台が再び全国に広がったこと，合同作戦，援軍といった複雑な戦争の登場（三国志IIの影響か？），他国の武将を引き抜くことができなくなった点などが挙げられる。武将の引き抜きは「戦国群雄伝」の面白さのひとつだったので残してほしかった気もするが，戦争のときにできる寝返りを促すコマンドで似たようなことは行える。むしろ，過剰な引き抜き合戦を抑えたと理解するべきだろう。

長宗我部親子奮戦記◆◆◆◆◆◆◆◆◆

シナリオ1「戦国の動乱」をプレイしてみた。武将は土佐の長宗我部氏。なんで土佐なんだといわれるとナニだが，まあ要するに気分で選んでみただけである。

いざ始めてみると全国制覇への道はなかなか険しそうだ。土佐は国力的にも平凡で，武将が4人と数がまるで足りない。長宗我部元親が優秀で軍師として活躍してくれるのと，隣接国が少ないのが数少ない利点といえば利点だ。

「武将風雲録」でのコマンド実行方法は変わっている。やはり今度も行動力によって出せるコマンドが決まるのだが，「戦国群雄伝」では武将ごとに行動力を持っていたのに対して，国ごとに行動力が設定されている。土佐の行動力はそこにいる武将全員が使えるわけだ。この行動力はその国の城主

長宗我部は土佐の大名。いかにして全国へ進出するか

の政治力で決まるので，政治力の高い武将を城主につけるとコマンドの数は増えることになる。

まずは土佐の国力の整備からとりかかる。開墾，治水，町への投資や技術革新を行った。国力がつくまでのプレイはわりと単調だが，結構いろんな演出があって楽しめる。たとえば軍師がいればアドバイスをしてくれたり，コマンドを実行しようとすると武将が現れて「ここは是非，拙者にお申し付け下され」と立候補（？）してきたりする。この願いを聞いてやると武将の忠誠度も上がって一石二鳥だ。開発のコマンドと，米の売買をうまく組み合わせて国力アップを図りながら，チャンスの到来を待つことにする。

やがて伊予に大内氏が攻め込んできて，伊予の大名河野氏が滅亡した。大内氏の兵力が大きいと見て，すかさず長宗我部元親を送り，大内氏と同盟を締結。狙いを阿波の三好義賢に定める。戦争の準備に鉄砲を手に入れたいのだが，今井宗久は「われわれも人の子ならば,金よりも心で動きます」といって首を縦にふらない。いまある鉄砲はわずか10。これでもないよりはと鉄砲部隊を作り，阿波に攻め入った。

「武将風雲録」の合戦には野戦，籠城戦，海戦の3つがある。攻め込まれたほうに選択権があるのだが，コンピュータは自分より多い兵力があると籠城戦，同等だと野戦を選ぶようだ。このときは野戦になった。結果は辛くも勝利。三好義賢は讃岐へ逃げていった。やはり鉄砲隊は頼もしい。遠距離から見る見るうちに相手を片づけていくのは，ほかの部隊ではできない芸当だ。

教養が低かったりすると品物を売ってくれない

城の戦闘画面

お茶を飲んで歌を詠む

次の標的は讃岐だ。手に入れた阿波も，やはり内政に力を入れる。土佐に香宗我部親泰などが新たに加わり，三好からぶんどった武将と合わせてなんとか数が揃った。長宗我部をプレイしていると，流しの武将（本当にそんなのがいる）だろうが何だろうがかまわず引き入れたくなる。

このときは幸運なことに，讃岐から次々と武将が兵を従えて投降してきた。どうも人事関係がおだやかでないらしい。兵力もなく武将もいない国など敵ではない。史上まれにみる楽勝で讃岐も手に入った。

ところが，ここからがなかなか大変。この一連の動きから伊予の大内氏は「長宗我部に四国統一の意思アリ」と見てとり，同盟を破棄してすかさず土佐に攻め込んできたのである。同盟を信用して兵力増強をさぼっていた土佐は総崩れ。辛くも守り抜いたものの，急に兵を増強せねばならなくなって，兵の質も国力も下がる一方。政情不安につれて，一揆は起こるし，三好氏は領地を取り戻しにくるし，不作になるしで現状維持が精一杯になってきた。やはり長宗我部が全国統一をするのは難しい……。

最新作の強み◆◆◆◆◆◆◆◆◆◆◆◆◆

まあこれはプレイした人がバカだったということでお許しいただくとして,「武将風雲録」全体のまとめに入ろう。

まずこのゲームの特徴である茶器と鉄砲だが，茶器のほうはどうもゲームに欠かせない要素としては働いていなかったという感がある。いますぐ茶会を開かなければならない理由というのがあまり見えない。

むしろ鉄砲のほうがゲームの特徴を際立たせるのに役立っていた。戦闘能力の低い武将でも，鉄砲を持っているとそれだけで脅威になる。これだけ魅力的な兵器なら，技術開発に精を出すのも悪くないという感じだ。鉄砲の有無が勝敗を分けるというのは時代考証としてもうなずけるし，またゲームとしても面白い。

野戦の画面。突撃だー

プレイしてみた感じでは，そういった新しい試みよりも，従来のシステムが整理され，エレガントになっていることの意味のほうが大きいと思う。また退屈にならないようイベントもいろいろ設定されている。技術を教えに鉄砲鍛冶が訪ねてきたり，金鉱脈の山師や富山の薬売りが来たりする。狩野山徳が来て，絵を描いていくなんてこともある。それに噂では，桶狭間の戦いや徳川家康の独立など，特別のイベントが設定されているという話だ。ひょっとすると本能寺の変まであるかもしれない。

画面もよくなったし，コンピュータの思考時間も短い。プレイするのにストレスはまったく感じない。そして，ここらへんが光栄のうまいところなのだが，プレイしてみると「いままでのシリーズのいいところを集めて作ってあるな」という感じがする。シリーズの前作からは微妙な手直しながら，どこかで確実によくなったと思わせる，その手法は絶妙だ。初めてこのシリーズをプレイする人にも，「戦国群雄伝」にはまっていた人にも，同じように「面白い」と思わせる，そういうゲームなのである。

どっしり王者の風格

よくも悪くも信長である。プレイすると確実に「よくなったな」と思わせるし，新鮮さも感じさせる。それでありながら，シリーズ一貫した感覚でプレイできる。前からの流れと新しさをミックスする技術にはすばらしいものがある。ただ，私はマウス操作がとってつけたようなものであるのと，音楽が「力を入れている」と宣伝しているわりにプアなのが気に入らないが，別に本質的な問題ではないだろう。

パソコンの素人は素人なりに，歴史マニアは歴史マニアなりにと，どんな人でも楽しめる，シミュレーション“お約束の”1本だ。

総合評価	0 5 10
操作性(キーボード)	★★★★★★★★
長くつきあえる度	★★★★★★★★
歴史マニア度	★★★★★★★★★
音楽	★★★★★
グラフィック	★★★★★★★
熱中度	★★★★★★★★★

みんなで作ろう，CPUの輪

Urakawa Hiroyuki
浦川 博之

ブロックを回転させて，重ねたり消したりというゲームは「テトリス」以降やまほど現れた。しかし，その中にはモノマネに終わらず，新鮮かつ面白いものも生まれている。「ループス」もそのうちのひとつだろう。

最近はけっこう面白いパズルゲームが多い。多いと私は思うのだが，どうもセールスのほうはかんばしくないようである。

まあ，たしかにパズルゲームを買うのはちょっと勇気がいる。せっかく必死に貯めたお金なんだから長く楽しめそうなRPGかなんかに使おうという感じだろうか。

しかし，ちょっと待ったと私はいいたい。RPGやADVってのは解けるまでは長いけど，いったん解けてしまうととたんに興味は失せてしまいがちだ。

それにプレイするのにも，パズルならちょっとした合間に軽い気持ちでホイホイっとできる。面白いものを選べば，意外に長くつきあえて，けっこうコストパフォーマンスが高かったりするもんなのだ。

わっかの作り方

ループスの内容をひと言でいうと，「出現するパネルをうまく配置して輪を作る」ということになる。しかし，まっすぐのパネルだったり，90度曲がったパネルだったりする。大きさもまちまち。パネルはテトリスの要領でくるくると回転。で，適当な居場所を探して置く。これの繰り返し。

パネルをつなぎあわせて輪にすると，得点が入って輪が消える。輪は長ければ長いほど，ややこしい形をしていればしているほど高得点。なあんだ，簡単じゃん。

かといって，安易に長いループを作っていると，手痛いお返しがやってくる。大きくて複雑な形をしたパネルがときどき出現するのだ。次のパネルはわからないので，出現したとたん，「置く場所がない！」という事態が発生することもある。制限時間内に置けなければ，哀れパネルは消滅してミス1。ミス3でゲームオーバーだ（ミスしていい数は画面左下に出ている）。

たまに，「ゴファー」と呼ばれる，菱形マークのパネルが現れることがある。これをパネルの上に置くと，そのパネルとそれにつながっているパネルがパッと消えるのだ。うまく使えば，地獄に仏である。

やりこんでいくと，このループスのどのへんにどんな駆け引きがあるかわかってくるわけだ。以下にまとめておこう。

1)　「このパネルが来ないとループが進まない」というような作り方を避ける
2)　特にクセのあるパネルは，それが来たときのための場所を取っておく
3)　使えないパネルは，ほかのループが伸びるのを妨げないように置く

で，出来のほどは？

気になるパズルゲームとしての出来は及第点に達しているといえる。プログラム的にもルール的にも目につく欠点はなく，わりとすぐに馴染める。パズルゲームとしてのツボは全部押さえてある。

2人プレイは共同？　それとも競争？

だが，プレイしていると何かもの足りない感じもしてしまう。「ああ，くやしい。もう1回だ！」と何度もやってしまうような魅力に欠ける気がするのだ。

「なんでかな」と考えてみると，どうも変化に乏しいためのようだ。テトリスだとレベル9ぐらいで1回ブロックの落下が速くなる「ヤマ」がある。クォースにもボスキャラやブロックの色が変わるというフィーチャーがある。それに比べるとひたすらパネルを置いていくだけで，展開が一本調子な感じがしてしまう（ただし，「DIFFICULT」モードでは速度が変化する）。

もうひとつ失敗点を挙げるとすれば，時間制限のゲージをパネルとは別に画面の下のほうに置いたことだろう。プレイヤーはパネルばかり見ていて，画面下の時間制限のゲージが見えない。ミスになっても，「あら，置けなかった」。あんまりプレイヤーの悔しさはかきたてられないようだ。

複雑な形になってしまうものだ

X68000用　5"2HD版　7,800円（税別）
ブロダーバンドジャパン　☎03(3341)1135

パズルゲームの定石

コンピュータ用パズルゲームとして，しっかり作ってあるのはたしか。2人同時プレイがあったり，パネルやフィールドのデザインを4種類から選べたりして，バリエーションも広く取ってある。パズルゲームの定石どおりに仕上げてあるといえよう。

しかし，回数をこなしてくるとパズルゲーム特有の麻薬性にやや欠けているなという感じがだんだん強くなってくる。パズルゲームの定石をこれだけきっちり押さえられるのだから，「楽しめるパズルゲーム」の定石も知らないはずはないと思う。決定的なインパクトに欠けるかもしれない。

総合評価（0〜10）

項目	評価
プログラム	★★★★★★★★
ゲームデザイン	★★★★★★★
操作性	★★★★★★★
グラフィック	★★★★★★★
演出	★★★★
熱中度	★★★★★★

スリルとロマンの星降る夜空

「3D 2（仮称）」を発売するという話題もあり，最近ノリにのっているM.N.M.Softwareから，ロマンチックなパズルゲームが発売された。重さの違う星屑を天秤の上にバランスを取りながらうまくのせるのだ。

Nishina Takashi
仁科 隆司

花火大会があったりなんかする季節。彼女と一緒に夜空を見上げている読者も多いことでありましょう。
女「きれいな夜空ね」
男「ほら，あれが北斗七星だよ（小学生でもわかるわい）」
女「あっ，流れ星！」
男「そういえば，こんな話知ってる？　この空のどこかに大きな天秤があって，落ちてくる流れ星を受けとめているんだって」
女「へえ，ロマンチックね。どうしてそんなこと知ってるの？」
男「M.N.M.のスターモビールをやってるからさ。今度家にやりにおいでよ」

この夏のくどき文句はこれで決まり。これで彼女を家に呼んだら，大外刈りから十字固めに持っていって寝技で一本だ。なんだかよくわかんないけど。

ふたつの天秤が同時に動く◆◆◆◆◆◆◆

でも当の本人が遊び方を知らないと恥ずかしいので，いまから予習しよう。

スターモビールはひと言でいって，「天秤のバランスを崩さないように星をのせる」ゲームである。ゲームが始まると，さっそく空からコンペイトウみたいな流れ星が落ちてくる。落ちてくる場所は自由に変えられるから，好きな皿の上にのっけてやる。ラウンドごとに決められた数だけ星をのっければ，ラウンドクリアさ（簡単簡単）。

だが，降ってくる星は，なんと色によって重さが違う（だんだんパズルの香りがしてきたぞ）。バランスを崩すとのっかっている星はぼろぼろと落っこちてしまう。落っこちるとRESTの数字がへっていき，0になるとゲームオーバーだ。

RESTを増やす方法もある。同じ色の星でほかの星をはさめばいい。はさまれた星は消えて，その分RESTが増えるようになっている。この「はさまれた星が消える」というのがポイントで，消えればその分だけ皿の重さが変わるわけだから，下手をすると天秤がひっくりかえって逆効果，なんてこともあるわけだ。覆星盆に返らず。

わかったかな。このゲーム，いったんわかると面白いけど，ルールを飲み込むには少し頭を使う。まあ，ただでさえ勉強しない夏休みだから，せめてスターモビールを理解するときぐらいは頭を使おう。

バランス感覚が大切なの◆◆◆◆◆◆◆◆

いやあ，結論からいってけっこう気に入りました。前作の「スライス」もお気に入りだったけど，こちらの出来もなかなか。

なんで気に入ったかというと，パズルゲームならではのジレンマやスリルをしっかり盛り込んであるからなんです。星が落ちてくる間に，どこに置くのがベストか考えださなきゃいけない。下手なところに置くと天秤全部がひっくりかえってしまう。どうしようどうしよう，というスリルが味わえます。よく考えると，どこにものせずに落としちゃうのが正解だったりしてね。

しかも，同じ星を9個積むとボーナスとか，天秤が水平な状態でラウンドクリアするとボーナスとか，特典を絶妙に配置してあるんです。ボクは何度ボーナスに惑わされて天秤をひっくりかえしたことか。逆にツツツツーっと天秤が動いてぴたっと止まったときのうれしさといったら，自分の車に駐車違反のチョークが引かれているのを発見したとき以上のものがあります。「あぶねーっ，ヒヤヒヤ」って感じで。

弱点はさっきもいったけど，ルールがとっつきづらいことかな。やったことのない人は「これ，何が目的のゲームなの？」とけげんな顔してのぞき込んでますからね。しかし，これはM.N.M.も承知しているようで，デモのときにちゃんとデモプレイつきでルールを説明してくれます。

さて，彼女のいない私はこのゲームを，熱帯夜の徹夜の友にでもするとしましょう。しくしく。

親切設計のゲーム説明デモ

星の重さを考えて。バランスが肝心

X68000用　5"2HD版	7,200円（税別）
M.N.M.Software	☎0423(60)3084

星がとっても重いから

ルールを把握するまでが大変かもしれないけど，わかるとサクサク遊べます。「本当かよー」などと疑わないように。要するに，理詰めで作ってあるというか，逆にいえばパズルゲームの文法をよく知っているなと感じさせるゲームなのであります。

コンペイトウがポロポロ落ちてくるオープニングもいい雰囲気出しているし，ステージが進むと星座のグラフィックが変わるというのもプレイに変化をつけてくれる。ゲームの性格づくりもうまいなということで，「まる！」の1本ですね。

総合評価	0　5　10
操作性	★★★★★★★★★
ビジュアル	★★★★★★★★
サウンド	★★★★★★
奥の深さ	★★★★★★★★★
ついもう1回度	★★★★★★★
熱中度	★★★★★★★★

生き残るためなら何でもやる

Akikawa Ryou
秋川 涼

「バトルチェス」のデザイナーたちが作ったというロールプレイングゲーム。無理やり放り込まれたバーガトリーというスラム街から脱出し，ディルムン諸島の平和を乱す邪悪な魔術師ナムターを倒すのが目的だ。

ボードゲームの世界でロールプレイングゲーム（以下，RPGと略）というものが登場したのは，結構昔のことだ。そして，そのRPGをパソコン上で再現しようとしたゲームが，かの「Wizardry」である。移行する際に，もともとのRPGが持つ「ストーリー性」は極力排除されたが，戦闘や成長などのシステムは必要不可欠な骨組みとして，しっかりと腰を下ろした。

それ以後，パソコンのRPGは進化を続けている。方向は源流への回帰，つまりボードゲームに近づくことである。

現在ではこうした流れによって，「ストーリー性」があり，パラメータが複雑というものが，RPGの中心になっている。「ドラゴンウォーズ」もそのひとつである。

どこへ行こうか

さて，ゲーム開始の鐘の音が(頭の中で)鳴ると，いきなり“身ぐるみ剝がされて，スラム街に放り出された”とのこと。

キャラクターの名前をクリックすると，ステータス画面になり，コマンドのアイコンも現れる。とりあえず，Itemのアイコンをクリックしてみると……，やっぱり何も持っていない。当たり前だな。

周りを見ると，見えるのは地平線と水たまりばかり。どちらへ行けばいいのだろうか。とりあえず歩き回るしかない。この“歩き回るしかない”という状態はゲーム開始直後だけではなく，ずっと続く。

歩き回る舞台は3D迷路。壁，海，岩，木などで多様化はされているものの，単なる迷宮だと思っていい。そのあちこちにイベントのあるマスが存在していて，そこに入ると突如グラフィックが表示される。これが「ストーリー」を構築している要素だ。店や病院なども同様の処理がなされる。

最初に笑わされたイベントは奴隷市場。“奴隷市場で自分を売るか？”というメッセージに答えさせられる。

奴隷になれば，持ち物を没収されるのは当然として，持ち物を見ると，“鎖”を装備している。状態も奴隷と表示されている。「奴隷のくせに……」とかいわれたりもする。ありがちだが，面白い演出だ。

目に見えるスキル値

パラメータが多いのもこのゲームの特徴。強さ，機敏さ，などという能力値はお馴染みだろうが，そのほかにスキルというのがある。超自然の知識，洞窟，森，山，町の知識，医術，登山，素手の戦闘，隠れ，鍵開け，スリ，泳ぎ，追跡，交渉，魔法，武器，と数多くの専門技術に対するスキル値が設定されている。

能力値やスキルは，レベルが上がったときに与えられるボーナスポイントを使って増加させる。

奴隷となり新たな場所へ

パラメータが多いのは結構だが，なるべくなら目に見えないところに存在して，自動的に増減するというのが，面倒臭くなくて，しかも，奥が深く感じられていいと思う。これは趣味の問題だろうけど，ボードゲームなどに慣れていない，いわゆる一般人の場合は同意見であろう。

動く敵

イベントが起こるとグラフィックで表示される。戦闘場面でも同様だが，そのときアニメーションする。敵によっては効果が出ているが，間抜けにしか見えないのもある。特に浮浪者，おいはぎ系のキャラクターは手つきがアヤシイ。また，魔法の効果なども視覚化すればよかったのにとも思う。

自分を売りたくはないが，あえて売ってみるか

X68000用　5"2HD版3枚組　9,800円(税別)
スタークラフト　☎03(3988)2988

あんまり先が見えない

マニュアルの最後に冒険者への助言が記されている。「このゲームでは，アイテム・スキル・魔法などを適切な場所で効果的に使わないと冒険を完了することはできません」「いろいろな出来事は1回しか起きないのが普通です。取り返しのつかないことをして，人々に嫌われないようにしてください」。一般的な話として，これはこれでいいと思う。しかし，それならば，適切な場所で効果的に使わなければ使わないなりに，取り返しのつかないことをしたならばしたなりに，ほかになんらかの解決法が存在しなければならない。好き勝手に行動しても許される，そういう自由度の高さがロールプレイングゲームという，いわゆる役割を演じるゲームには必要だと思う。このゲームはまだ最後までやったわけではないが，いまのところは好き勝手にやってもそれなりに進んでいるようだ。

「とっつきはいいけど，そのあとはひきずりこむようなストーリー展開ではない」というのは，賛否両論になるところだろう。

総合評価	0　5　10
ストーリー	★★★★★★★
システム	★★★★★★★★
グラフィック	★★★★★★
効果音	★★★★★★
パラメータの多さ	★★★★★

優勝しようよ

Mounai Toshiyuki
毛内 俊行

いいスコアを出すためには,正確なショットが必要だが,それだけではだめ。的確な戦略,攻め方というのが重要になってくる。エイトレイクスは非常に(非情に)難しいコースだが,優勝することは決して不可能ではない。

このコースで恐いのは湖へのウォーターハザードとOB。優勝を狙うには,どうしてもこの2つとの縁を断ち切らなければいけない。特に,インに入ってからはパーをキープすればよし。それ以上を望めば必ずスコアを崩してしまうだろう。

そこで,今回は正確なショットが要求されるインコースの中から,13,15,18の3ホールを紹介しよう。

●Hole13 Par3 195yards

フェアウェイに木が覆いかぶさっていて,極端にせまいコース。しかも,グリーンは湖にはりだしていて,湖へ向かって傾斜している。無造作にティーショットを打とうものなら,ボールは木の枝にひっかかるか,湖へ向かって,するするする,ぽちゃん。

攻め方は2つ。まずひとつは,トップスピンをかけて低い弾道でグリーンを狙う方法。ただし,極端に弾道が低くないと,ボールが木にひっかかってしまうので1Wを使う。ということは当然,ボールを打つパワーも控えめに。力の加減を1歩間違えれば,ボールは早いグリーンをコロコロと湖へ。あまり上手な攻め方ではない。

2つめの攻め方は,バックスピンをかけて,じゃまな木の上空からグリーンを狙う方法。ただし,直接狙うのは危険。グリーンでワンバウンドしたボールは必ず,バンカーか湖へ向かうからである。3Iでグリーン直前のフェアウェイにボールを落とすのがいい。20～30ヤードのアプローチショットが正確に打てるならパーセーブは堅い。ただし,高い弾道を打つため,強い横風が吹いているときは苦労するだろう。

この木々がじゃまになる

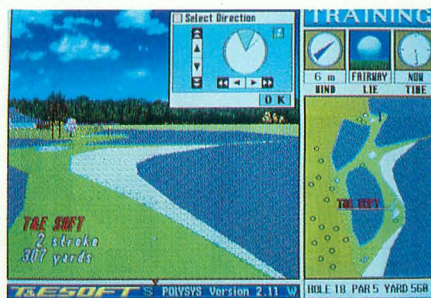
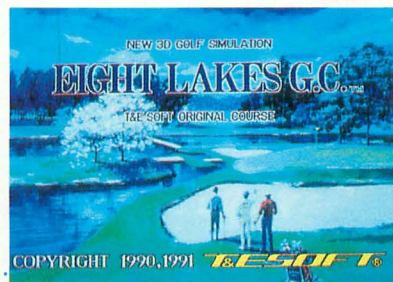
●Hole15 Par3 160yards

断崖絶壁の向こうにグリーンの見えるショートホール。グリーンはせまく手前は深い谷,向こうはクリークにはさまれ,また起伏が激しいため,グリーン上でバウンドしたボールがどこへ行くかはまるでわからない。ここでスコアを崩した記憶は一度や二度ではない。神や仏に祈りたくなる。

ティーショットは4Iか,3Iでバックスピンをかけるのが基本。スタンスをクローズに構え,ドローボールでグリーンの左を狙って打つのがいい。グリーンの左にはバンカーやラフなど,ボールが転がるのを防ぐ要素が多いからだ。ここでバンカーを恐れてはいけない。バンカーならペナルティはないが,OBは1ペナルティなのだ。

セカンドショットはピンの手前に落とす感じで,少し弱めに叩こう。力加減を間違えると,グリーンを越えて谷底へ……。

●Hole18 Par5 570yards

このホールはフェアウェイこそせまいが,確実にパーが狙えるホール。ランキングが1位だったら,無理をせずパーを狙おう。ティーショットは3W,セカンドショットは4Wでそれぞれバンカーの左を狙えば,湖にボールを落とすことなく3オンできる。

問題は,このホールで確実にバーディ,もしくはイーグルがほしいときだ。基本的に570ヤードもの距離を2オンさせるのは無理がある。が,場合によっては不可能なことではない。そう,風である。このホールでは,ティーグラウンドからグリーン方向に風速5,6メートル(もしくはそれ以上)の風が吹いていたなら,チャンスである。

そのときは思い切って池越えを狙っていこう。ティーショットはもちろんドライバー。ボールの着地点がせまいので,バックスピンをかける(ただしテンプラにならないように気をつけて)。ボールを打つ方角は,方向指定のとき,左に小さく8回(大きく1回)回した位置あたりがいいだろう。

セカンドショットは,風と距離に相談しなくてはいけない。使うクラブは3W。距離が230ヤードを切るようなら2オン可能であるから,グリーンを狙おう。250ヤードくらい残っているのなら,どんなにフォローが強くてもグリーンに届かないので,グリーン手前のフェアウェイに手堅くのせていこう。第3打は確実にピンを狙えるので,距離に応じて臨機応変に対処すればいい。

断崖の向こうにグリーンが見える

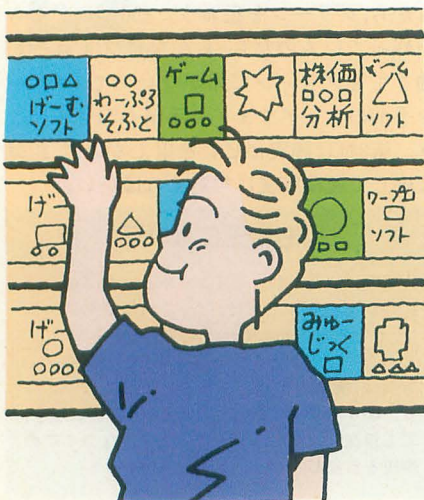
せまくて変な形のフェアウェイ

君も帝王になれる

さて,2カ月にわたってエイトレイクスの攻略について説明したわけだが,これはほんの一例で,もちろん人それぞれの攻略法があるだろう。このコースではアンダーパーでホールアウトできれば,優勝の可能性がある。70がひとつの目安。68が出せれば8割以上の確率で優勝だろう。君も精進して,J・ニクラウスのようなゴルフの帝王を目指してみたらどうだろうか。

X68000用 5"2HD版2枚組 5,800円(税別)
ティーアンドイーソフト ☎052(773)7770

AFTER REVIEW

とってもかわいらしく，メルヘンな雰囲気にだまされてはいけません。「メルヘンメイズ」はジャンプなどに結構テクニックが必要で，なおかつ楽しいというアクションゲームの王道を行く作品なのです。

メルヘンメイズ

▶アリス以上に大きな声で，「あん！ あん！」といっている，あんどうみちこ（はたから見るとばか)。安藤　道子(18)宮崎県

▶要2Mバイトだけのことはある。
箕浦　真(20)東京都

▶アーケード時代から好きだったけど，すぐに消えちゃって，ちょっとくやしいと思ってたから。　鈴木　茂(20)東京都

▶かわいいキャラ。変な死に方。
山岸　稔(18)埼玉県

▶福原徹さんにひと言。メルヘンメイズはマイナーなゲームじゃないぞ。あと，README.DOCが面白かった。
勝又　基至(17)静岡県

▶メルヘンタッチなわりに，むずいぞー。
荒木　幸博(21)宮城県

▶ゲームセンターでクリアできなくて残念に思っていたら，X68000で出たから。
細江　昭史(18)長野県

▶最後の女王はなかなか強かった。
森　健一(18)千葉県

▶私はSPSに対する恨み(イシターのとき)を忘れ，浄化することができた。このソフトは買いです。いまのSPSは昔とはえらい違いだ。　森崎　剛(18)広島県

▶誰にでもとっつきやすい内容。
清瀬　亮治(18)広島県

▶俺のジョイスティックは斜めにレバーが入らないから，面白いこと面白いこと。
小原　健一(17)宮城県

▶対高所恐怖症ソフトになるんじゃないかと……。キャラもかわいいし。
厚井　浩一(19)広島県

▶とにかく「あんっ」だ，「あんっ」。ろりこんじゃないぞ。　柿沼　光(17)埼玉県

▶「メルヘンメイズ」はゲームセンターではやったことがない。雑誌の紹介記事を読んで，存在は知っていたけど，なんとなく面白そうだなあ，ぐらいにしか思っていなかった。しかし，X68000に移植されると聞いたとたんにやりたくなってきた。そして，買ってきてゲームを見てみると……，キャラクタのかわいさが中心のゲームだと思ったら大間違い。ジャンプは苛酷だし，敵はしつこいわで，なかなかの手応え。もとを知らないので，はっきりしたことはわからないけど，たぶんほとんど完璧な移植なんでしょうね。さすが，SPSさん。ところで，このゲームって，アクション？　それとも，シューティング？　まあ，どっちでもいいか，そんなこと。　須田　昌章(19)島根県

▶主人公のアリスはもちろんのこと，登場する敵キャラクタの数々が気に入った。どれもこれもかわいいというか，なかなかいいデザインである。ペンギンに，トランプの兵隊，タモリを弁髪にしたような尺取り虫，そして，とてもカタくてやっつけると

▲柿沼　光(17)埼玉県

▲佐田 匠 千葉県

カマボコ(?)を切ったように死ぬ，ダンゴ虫のようなキャラ。うーん，みんな生きている。　駒田 完二(18)岐阜県

▶最後の女王様はズルをせずに勝ったことがない。つまり，いつも女王の右横に隠れて，敵キャラが画面に出ないようにして戦っている。まともに敵や弾を避けながら倒すのは至難の技だなあ。

森島 和幸(28)奈良県

▶原作がまずとんでもない。「不思議の国のアリス」の世界がこういうかたちで，こういうゲームになっているのは芸術的次元で凄いことだと思う。企画倒れにならず，完璧に再構成されたゲーム世界。ハードウェアの限界など意に介さぬといった感じのキャラクタデザイン。ゲームは次第にテンションを上げつつ，シューティングゲームから格闘ゲームに移行していく。群れをなして迫りくるトランプ兵……。妥協というものを知らんのか？

そっくりそのまま持ってきた移植も凄い。え゛，こんなに色数が出たっけ？ “うさぎさんばりあー”もちらつかない……。

両者に感じられるのは「情熱」だ。なにかよくわからないが，訴えるものがある。そーか，ならジョイスティックを壊すくらい熱中しても不思議はないな。　(S.N.)

▲澤田 良満 愛知県

▲山辺 由紀子(18)長崎県

発売中のソフト

★イース　電波新聞社
X68000用　5″2HD版2枚組 9,600円(税別)

★生中継68　コナミ
X68000用　5″2HD版2枚組 9,800円(税別)

★ジーザスII　エニックス
X68000用　5″2HD版 8,800円(税別)

★グループ・エックス　コムパック
X68000用　5″2HD版2枚組 7,800円(税別)

新作情報

★ボナンザブラザーズ　シャープ
X68000用　5″2HD版 価格未定

★オルテウスII　ブラザー工業(TAKERU)
X68000用　5″2HD版 4,800円(税込)

★アクアレス　エクザクト
X68000用　5″2HD版2枚組 8,700円(税別)

★機動戦士ガンダムクラシックオペレーション　ブラザー工業(TAKERU)
X68000用　5″2HD版 7,100円(税込)

★キャメルトライ　電波新聞社
X68000用　5″2HD版 価格未定

★3D 2（仮題）　M.N.M.ソフトウェア
X68000用　5″2HD版 7,200円(税別)

★フューチャーウォーズ　スタークラフト
X68000用　5″2HD版 9,800円(税別)

★マジックキャンドル　スタークラフト
X68000用　5″2HD版 9,800円(税別)

★ダーウィンズジレンマ　スタークラフト
X68000用　5″2HD版 9,800円(税別)

★eXOn　日本ソフテック
X68000用　5″2HD版 価格未定

★インペリアルフォース　システムソフト
X68000用　5″2HD版 価格未定

★ロードス島戦記　ハミングバードソフト
X68000用　5″2HD版3枚組 9,800円(税別)

★大戦略III '90　システムソフト
X68000用　5″2HD版2枚組 9,800円(税別)

★銀河英雄伝説IIDX+kit　ボーステック
X68000用　5″2HD版 5,000円(税別)

★シュヴァルツシルトII　工画堂スタジオ
X68000用　5″2HD版2枚組 9,800円(税別)

★F15ストライクイーグルII　マイクロプローズジャパン
X68000用　5″2HD版 価格未定

★フェアリーランドストーリー　SPS
X68000用　5″2HD版 価格未定

★スーパー上海ドラゴンズアイ　ブラザー工業(TAKERU)
X68000用　5″2HD版 7,800円(税込)

★SPINDIZZY II　アルシスソフトウェア
X68000用　5″2HD版 価格未定

★ドラッケン　エピック・ソニー
X68000用　5″2HD版 9,700円(税別)

★シムアース　イマジニア
X68000用　5″2HD版 12,800円(税別)

★パワーモンガー　イマジニア
X68000用　5″2HD版 12,800円(税別)

定番MIDIソフトの最新作登場

Misawa Kazuhiko
三沢 和彦

Musicstudio，Mu-1両シリーズの最上位バージョンが登場。ステップ入力を強化したMu-1 Superだ。早くもSC-55やスタンダードMIDIファイル(3.5インチFDD対応)をサポートした意欲作。LA音源音色エディタも付属している。

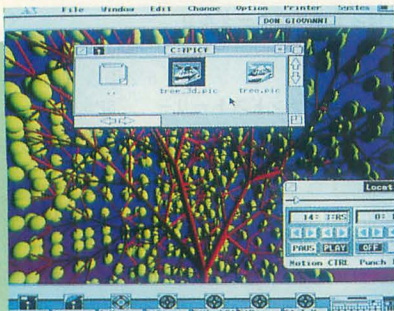
X68000のMIDI環境をサポートするソフトウェアとして定評のあるMusicstudioの最高級バージョンが手元に届きました。その名も「Musicstudio Mu-1 Super」。これまでにサン・ミュージカル・サービスやシャープから発売されているMusicstudio ver.2.0とMu-1の機能をすべてあわせ，新しい機能を付け加えたという究極のMTR（マルチトラックレコーダ）ソフトです。

また，このソフトウェアの評価版と一緒に音源モジュールの新製品，サウンドキャンバスSC-55も試してみる機会がありました。この新モジュールは，これまたDTM（デスクトップミュージック）の基本的な音源モジュールとして不動の地位を築いたMT-32，CM-64/32に続いて，同じローランドから発売されたものです。ここでは，これら最新のソフトウェアとハードウェアとの組み合わせによる最新鋭のDTM環境をテストしてみたいと思います。

Mu-1 Superの概観

Mu-1 Superの特徴は，

1) MTRとしての基本機能を完成させた
2) CM-64/32を最大限使いこなすためのユーティリティを充実させた
3) GS規格対応

の3つにあるといえましょう。

X68000用	5"2HD2枚組	39,800円(税別)
サン・ミュージカル・サービス		☎03(3419)8839

まず，1)のMTRとしての機能ですが，内容的にはMusicstudio ver.2.0と同じです。といっても，このSuperはMu-1のアップグレードバージョンなので，オリジナルのMu-1との違いを中心に述べます。

まずは，Musicstudio ver.2.0同様，内蔵FM音源もAD PCMもドライブでき，X68000のまわりのあらゆる音源を同期演奏させることができるようになりました。

MTR画面上で各トラックのレベルメーターの下にMTR，FM，およびPCMの表示が出るので，それぞれのトラックがどの音源をドライブしているかがひと目でわかります。

もともとMu-1ではステップ入力，ステップエディットには対応していましたが，初心者MIDIユーザーにはリアルタイム録音は必要ないとの判断からか，外部MIDI楽器で演奏したデータを取り込むリアルタイム録音はサポートされていませんでした。しかし，いわゆるMTRではこのリアルタイム録音こそがもっとも基本的な機能です。もちろん，Musicstudio ver.2.0には元からあったものです。これが今回のSuperではサポートされています。

コンピュータミュージックではステップ入力が基本だという意見もあり，本当にリアルタイム録音は必要な機能かどうかは意見の分かれるところですが，ないよりはあったほうがよいに決まっていますので，オリジナルMu-1ユーザーには今回のバージョンアップは朗報といえましょう。

さらに，ステップエディットについてもUNDO機能が追加されるなど，より操作性が向上しています。

CM-64/32対応の機能

さて，もともとMusicstudio ver.2.0のユーザーだった人にはなんの恩恵もないのかといいますと，そうでもありません。オリジナルのMu-1にはミュージ郎，ミュージ君のデータをMu-1用に変換するデータコンバータが標準でついていました。これで，すでに演奏ライブラリの充実しているPC-9801シリーズの演奏データを丸ごと利用することができ，とても経済的です。でも，このデータコンバータはあまり本質的なものではないかもしれません。

そこで，これまでのどのソフトウェアにもなかった新機能をいくつか紹介しましょう。そのひとつが，2)のCM-64/32対応ユーティリティのことです。まずメインのMTRを下のほうにスクロールさせていくと，52音ドラムミキサー，LA音源エディタ，およびグラフィックキャンバスの3機能が切り換えられるウィンドウが現れます。

52音ドラムミキサーはCM-64/32，MT-32のドラムパートの音色を自由に入れ替えるユーティリティです。これまで，ドラムパートの音色は鍵盤上の音階1音ずつにそれぞれ1種類のドラム音を割り当てていました。プログラムチェンジでは音色の差し換えができないため，メインのMTRウィンドウ上ではリアルタイムにドラム音を変えることができなかったのです。

たとえば，スネアドラムの音をアコースティックスネア（D1音）とエレクトリックスネア（E1音）のどちらにするかを実際に曲を聴きながら切り換えて決めたいというとき，いままではドラムパートのトラックの演奏データをいちいち書き換えなければなりませんでした。

今回のドラムミキサーでは，画面上にドラム音名が並んでいて，それぞれの音名に実際のドラム音のテーブルナンバーを割り当てていけばよく，しかも，それらをマウス操作で切り換えていくことができます。ですから，最初は曲データをアコースティックスネアで書いておいて，あとは実際に曲を流しながらアコースティックスネアの設定音をエレクトリックスネアのナンバーに切り替えていくだけで一発変換OKというわけです。

加えて，各ドラム音の音量を独立に変え

●旧Mu-1ユーザーはもちろん，シャープから発売のMusicstudio PRO-68Kのユーザーにもu-1 Superへのバージョンアップサービスがされます。登録ユーザーにはシャープからの案内状が届くはずですので，その指示に従ってください。

るボリュームスライダーもあり，全体のバランスを好みに従って変えていくこともできるようになりました。

LA音源エディタはCM-64/32, MT-32の持つメモリ領域にユーザーが自由に音色データを設定するためのエディタです。音色データのやりとりはMIDIエクスクルーシブで転送するのですが，ローランドのエクスクルーシブデータフォーマットは複雑で，上級者でなければ使いきれません。そこで以前私自身がこのフォーマットを解析してX-BASIC上でLA音源エディタを組み，Oh!X誌上で発表したこともありました。

このたびMu-1 Superについてきた LA音源エディタの基本機能は私の作ったものと同じですが，ウィンドウデザインを始め，操作性はさすがに格段よいものになっています。LA音源のエディットのコツについてはここで説明し始めると，LA音源の仕組みから始めなければならないので割愛します(1990年3月号参照)。

GS規格対応

最後に今回の目玉ともいえる機能を説明しましょう。3)のGS規格対応です。メインのMTRウィンドウはこのGS規格に対応しています。GS規格の主旨は次のとおり。

1) MIDIの16チャンネルをフルにサポートする16パートのマルチティンバー
2) 最低24音の最大同時発音数を保証する
3) プログラムチェンジを0〜127のバンク切り換えとし，標準音色をバンク0に設定
4) ビブラート，パン，リバーブなどのコントロールチェンジを充実させる

機能的に新しいのは2)のバンク切り換え式プログラムチェンジで，これは1台のモジュールで128×128＝16384種類の音色が扱えるということになります。従来のプログラムチェンジに比べてかなり容量に余裕ができたので，バンク0は音色とプログラムナンバーとの対応を標準にし，演奏データが器材を選ばないようにしようというのです。

では実際にMu-1 SuperではこれらのGS規格にどう対応しているのでしょうか。

1) プログラムチェンジのためのバンク切り換えが画面上で操作できる

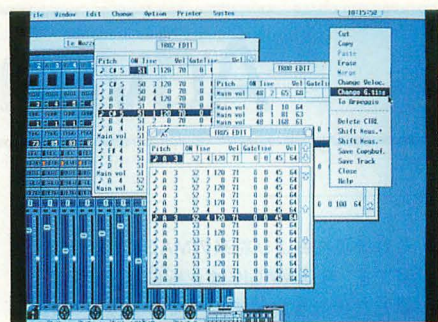
メインのMTRウィンドウでMIDIチャンネルやプログラムチェンジの設定に加えてバンクの表示が追加されています。たとえ演奏中でもここの設定を変えるとバンク切り換えに伴って同じプログラムナンバーでも音色が変わっていきます。

2) ステップ入力でバージョンアップしたコントロールチェンジも設定できる

よく使うのは，エフェクトの設定でしょう。また，バンク切り換えも画面上の操作だけでなく，コントロールチェンジのかたちでデータをセットすることも可能です。

GS規格については本誌でも何度か解説されたことがあります。ちゃんとしたMIDI規格の一部となることを願っていましたが，残念ながら国内のMIDI協議会では正式採用されていないようです。それに伴ってか，従来「GSスタンダード」として知られていたこの規格は「GS規格」というローカルなものになってしまいました。コンピュータを使っている場合，GS規格はありがたい内容ですので，今後は実力で実質的標準となるようにがんばってもらいたいものです。

次のバージョンアップは?

上で述べたように，Mu-1 SuperはGS規格の基本機能には対応させましたが，あくまでも基本機能だけで，このままではSC-55そのものを使いこなすまでにはいきません。もちろん規格を最低限満たしているという点では，いまのままでも十分合格といえましょう。ただ，GS規格を普及させていくためにはその1号機であるSC-55をまず

改良されたステップエディット

は普及させなければならないはずでしょう。そのためには，SC-55の大きな特徴である音色エディットについても操作性を上げておきたいところです。ちょうど，今回付属しているLA音源エディタに対応する音源エディタの開発が待たれます。

サン・ミュージカル・サービスのソフトは元のデキがよいので，いつもほめ言葉が多くなってしまいます。最後にただひとつ大きな注文をするとすれば，それはSX-WINDOW対応バージョンを出してほしいということです。独自のミュージックシェルもよいのですが，統一された環境でできたほうがいいに決まっていますからね。

SC-55試用記

SC-55ではGS規格で設けられた128種類のバンクをフルに使用しているわけではなく，バンク0の標準音色に128種類に加え，標準音色の一部の音色に計59種類のバリエーションを設定しています。これだけの音色数があればごく一般のコンピュータミュージックには十分すぎるほどのモジュールといえましょう。

実際の音は，CM-64のRS-PCMを基に作っているようで(音源方式は明示されていない)，LA音源よりはいい音をしています。ただ，かなりマイルドな音の感じが強く，いわゆる「デジタル楽器の音」に慣れている人よりも，昔ながらのアナログシンセサイザ好みの人のほうにウケるのではないかと思います。

さて，SC-55は音源モジュールであってシンセサイザではないので，音色をまったく新しくエディットすることはできません。しかし，各パートごとにリバーブおよびコーラスのエフェクトを独立にかけることができます。リバーブはコンサートホールや室内の残響音を加え，コーラスは微妙に高さの異なる音を混ぜて，音に厚みをつけます。これらはかなり強力で，エフェクタなしでは情けない音もエフェクトをかけるとかなり納得のいく音に仕上がります。

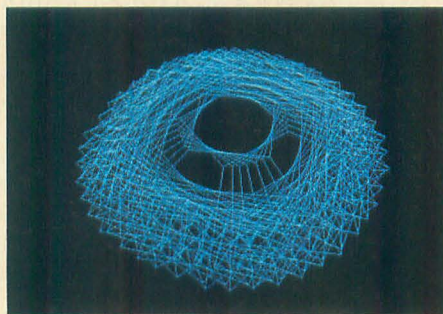
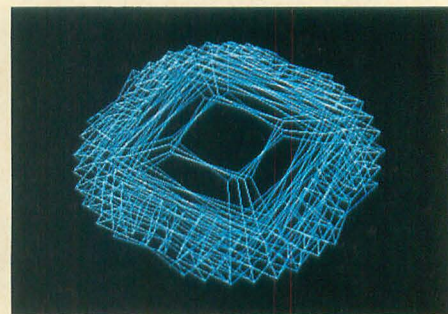
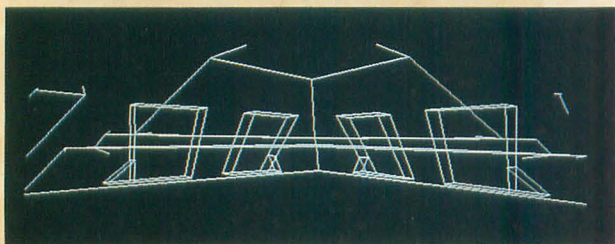
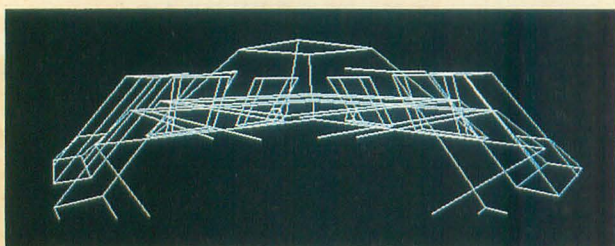
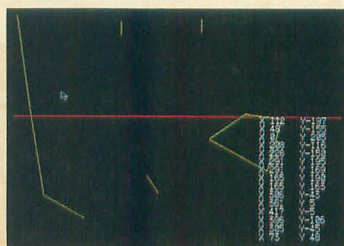
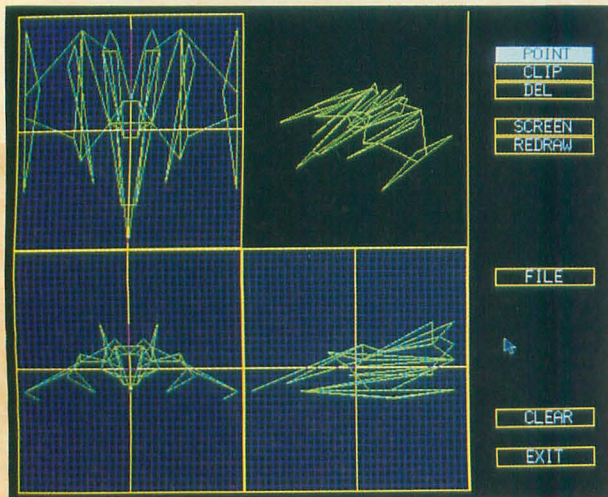
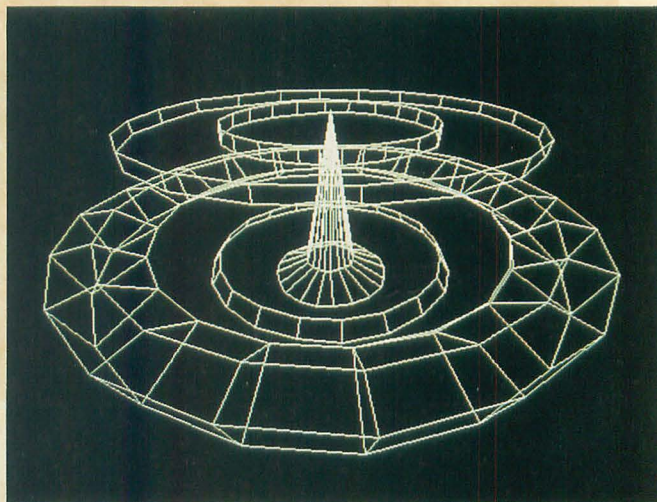
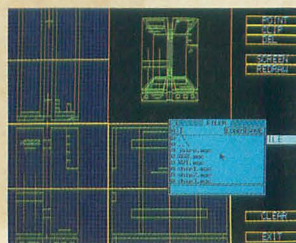
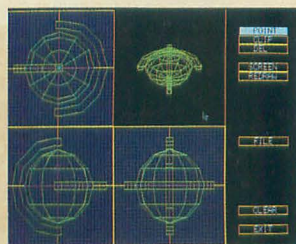
さらにそれだけではなく，各パートごとにビブラート，カットオフ，レゾナンス，エンベロープといったエディットがかけられます。ビブラートは音の高さを震わせる効果です。カットオフは音の倍音成分の調整で，倍音成分を多く含ませると金属音のようなとがった音，反対に少なくすると柔らかいが単調な音に近づきます。レゾナンスは，カットオフでカットする倍音成分を強調させる効果で，レゾナンスを大きくするほど癖のある音になります。エンベロープは音の強さの時間的変化で，音の立ち上がりの速さや減衰の速さを調整します。

いま述べたビブラート，カットオフ，レゾナンス，エンベロープは初期のアナログシンセサイザの基本的な音色構成で，実際，これだけのパラメータが変えられるのはシンセサイザそのものとほぼ同じ機能を持つことになります。

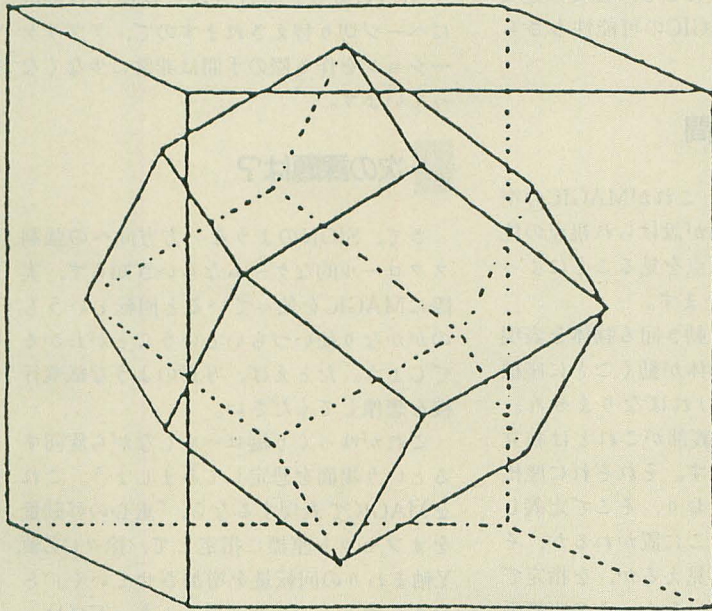
これだけ豊富な機能を搭載したSC-55ですが，機能の豊富さに比べて，パネル面のスイッチ類の数が少ないために操作が複雑ですぐにはマスターできません。また，ディスプレイも限られているために設定内容の確認もわかりやすいものとはいえません。さらに，本体にメモリがないので，一度設定した内容は電源を切るとなくなってしまいます。しかし，すべてのパラメータはMIDIを通して外部から設定可能になっています。特に，コントロールチェンジの情報としてCMシリーズにあったモジュレーション，パンポットといった基本的なものだけでなく，各パートごとのエフェクトおよび音色エディットも制御できるようになっています。

これらの情報はいままでのCM，MTシリーズではエクスクルーシブで情報を転送していたため，情報も重く，しかも汎用性がない状態でしたが，これからはコントロールチェンジでトラック別により柔軟に設定がかえられるようになりました。こうしたことから，このSC-55はコンピュータ画面上で操作するのがもっとも効果的な使い方と思われます。演奏データの先頭に音色エディット，エフェクタなどの設定を置いておけば，常に同じように演奏データを再現させることができるのです。

Advanced Magical World

これは現在および将来のMAGICのたどる姿です。今後MAGICはさらに高速化され，新たな機能を加えていくでしょう。それを支援するためにはさまざまなツールも必要となります。ここに挙げられたものはMAGICの世界を構成する最初のプログラム群なのです。少しでもこれからMAGICの開く世界を感じてもらえたでしょうか。

新しいMAGICの速度比較例

旧MAGICだとこれくらい

Z軸のクリッピング例。5月号の付録ディスクからTIRREL.Xを立ち上げて，Zキーを押し続けたところ

回転体専用のモデラー。左のように指定しておくと，下のような回転体を自動的に作成する（分割数20）。このプログラムはMAGIC.FNCを使用して制作されたものだ

開発中のMAGICモデラー。ごくオーソドックスな構成の物体エディットツール。完成したら配布する予定

ポリゴン版MAGIC（参考）の表示例

[特集]

Brush up your MAGIC

汗を流せば速くなる。知恵を尽くせば速くなる。信じていれば速くなる。単に動かすだけのプログラムと，磨かれたプログラムとでは本質的な差が存在する。ことソフトウェアに関しては限界は必ず打ち砕かれる。そして，それを信じることが次の限界を打ち破っていく。
我々の前には作り上げることのできる世界がある。共有できる知識がある。それらを認識できない人はパーソナルコンピュータを理解しているとはいえないだろう。多くの人の協力を得て，いまMAGICは第1段階のバージョンアップを終えた。いや，位置づけとしては，これまでのMAGICはまだMAGICではなかったといったほうがいいのかもしれない。
そしてMAGICの世界をもっと具体的に展開することが可能になった。テクニックからイマジネーションへフェイズシフトが行われていく。次はMAGICに広がりを持たせることが重要なのだ。これにはより多くの人々の参加が必要になるだろう。もともとMAGICはプロのテクニックを手軽に扱えるように用意されたパッケージだった。それを使いこなすのに必要なものは技術ではなく想像力なのだ。夢見たことはないだろうか？　自在に扱えるリアルタイム3Dグラフィックを……。
そして，いままたMAGIC ver.3.0への道も始まろうとしているのだ。

CONTENTS

MAGICの原理と活用

Nakano Shuichi 中野 修一

新しいMAGIC。ここではMAGICの基本的な考え方と，これからMAGICのたどるであろう道程を概観してみましょう。これからのMAGICはどうあるべきか，どうなっていくのかをひと足先に解説しておきます。

手軽に3Dグラフィックなどを楽しむことができるリアルタイムグラフィックパッケージMAGIC。MAGICでは3Dグラフィックを簡単かつ高速処理するために，いくつかの約束事があります。それをきちんと守れば誰でも比較的簡単にMAGICの恩恵を授かることができるのです。

初めに3D部分以外のMAGICの基本的な考え方を示しておきましょう。

MAGICはメモリ上にあるデータを読み込み逐次実行します。メモリに置かれたデータをすべてコマンド列と解釈します。基本的にMAGICはデータに間違いがあることを考慮していません。どんな情報があっても勝手に解釈しますので，バッファのアドレスやデータの内容は絶対に間違えてはいけません。

具体的にはデータエリア先頭から1ワード（16ビット）のデータを読み込み，それをコマンドワードだと解釈します。コマンドワードの内容は71ページのとおりです。それぞれのコマンドによって続くデータをパラメータと解釈して実行します。パラメータの内容も71ページのとおりです。これにちゃんと従うだけですから，MAGICを使うのは特に難しい技術は必要ありません。

付録ディスクに収録されたSIONは68000のマシン語を始めたばかりの山田氏が作成したものです。MAGIC自体も描画関係にIOCSをそのまま使うという，“MAGIC”としては邪道なものでした。それでもSION程度のゲームなら比較的簡単に作れてしまうのです。SIONはMAGICのデモとして作成されたものですので，あえて512×512のモードで作成されています。MAGICの3D処理のほとんどは描画時間ですから，256×256モードにするだけで物理的に高速化する可能性を残しています。

今回バージョンアップしたことにより，一応“MAGIC”の名に恥じない程度の速度にはなりました。MAGICの可能性もさらに広がっていきます。

MAGICの空間

図1を見てください。これがMAGICの空間です。3次元の座標軸が設けられ視点の位置は固定され，常に原点を見ることによって3D処理を軽減しています。

この座標系だけでは動き回る物体を表現するのは大変です。物体が動くごとに座標をすべて書き換えなければなりません。MAGICでは物体の定義部がこれとは独立した空間となっています。それぞれに座標軸と原点を与えられており，そこで定義した物体が全体空間のどこに置かれるか，その物体はどんな角度で見えるか，を指定できます。物体の移動だけなら，その座標系の位置を示す座標を書き換えるだけでいいわけです。

全体空間と物体空間に分離されていることにより，個々の物体単位での運動が格段に指定しやすくなります。MAGICでたくさんのオブジェクトを扱う際には，メモリ上にいくつかの3Dデータを置いておき，それぞれで全体空間に占める位置，回転角度を指定しておきます。あとは表示物体リストに表示したい物体データの先頭アドレスを登録してAUTO命令を実行するだけです。

AUTO命令で実行された物体は自動的にページ切り替えされますので，アプリケーションを作る際の手間は非常に少なくなっています。

次の課題は？

さて，SIONのような一方方向への強制スクロール的なゲームならいざ知らず，実際にMAGICを使っていると回転というものがかなり使いづらいということがわかるでしょう。たとえば，写真のような紙飛行機を想像してください。

これがゆっくり縦ロールしながら旋回するという場面を想定してみましょう。これをMAGICで表現するなら，「重心の移動量をオフセット座標に指定して，徐々にZ軸，Y軸まわりの回転量を増加させていく」となると考えるのが普通でしょう（写真1）。しかし，これを実際にコーディングすると紙飛行機は写真2のような挙動をします。

これは座標を1軸ずつ回転しているからです。回転は行列の一次変換で表されるということは，

$$\begin{bmatrix} X' \\ Y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix}$$

で座標をひとつずつ新しい座標に変換しているわけです。ところが，2軸以上の回転となると，この変換された座標を基に同様の操作を繰り返していきます。つまり，最初の座標とは違うものを元の座標軸で回転させているのです。ですから，だんだんずれていくのも当然といえます。さらに，納得できない人もいるかもしれませんが，回転

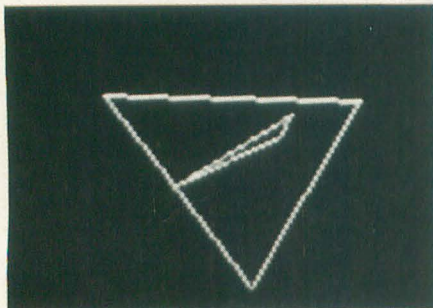



写真1　旋回する

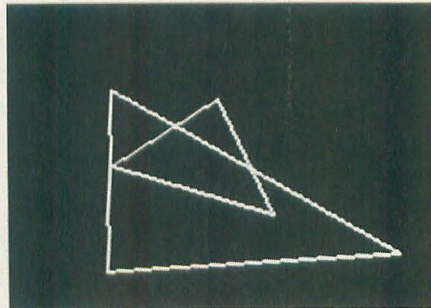



写真2　変な方向を向いてしまう

の順番が違えば紙飛行機の最終的な方向は少し異なります。

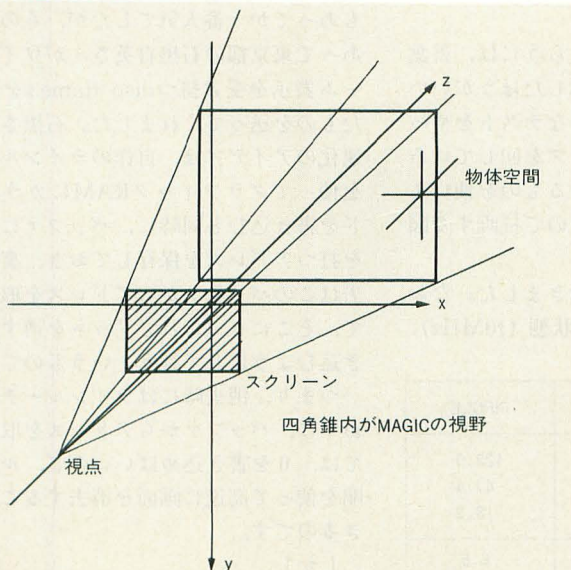
おそらく，アクションゲームの最中ならそれほど気にならない程度のものかもしれません。しかし，大変なのはそこから急上昇や急降下をしようとしたときです。今度は，機体が思いもよらぬ方向に向いてしまいます。最初に機首をZ軸にピッタリあわせていても，回転の最中に機首はZ軸とはまるで違った方向を向くようになってしまうのです。思いどおりの挙動をさせるのは至難の業といっていいでしょう。

これを解決するにはオイラー角というものを導入しなければなりません。

なにが悪いのかというと，回転後の座標軸とデータの関係が変わってしまうことです。回転は常に物体空間内で行われています。それゆえ，物体がどんなに回転しようと全体空間の座標軸とデータ空間の座標軸は常に平行を保っています。

ではデータを回転させたときに座標軸も回転させればいいのでしょうか？ いいえ，中身のデータ自体は回転させずに座標軸だけを回転させればよいのです。

このような方法を取った場合の，観測者のいる大きな空間の座標軸と物体1つひとつの座標軸の取る角度，それがオイラー角と呼ばれています。物体を運動させるときには，もはやデータそれぞれを扱うのではなく，データの入った空間をまるごと運動させるのだといってもいいでしょう。これなら物体が変な向きを向くことはありません。

具体的なオイラー角からの座標算出方法は，以前，祝一平氏がC調言語講座で作成したフライトシミュレータの解説を参考にしてください。

なお，フライトシミュレータにオイラー角の導入が絶対に必要か？ というとそうでもありません。MAGIC同様の座標系でまかなっている例も多々あります。しかし，次の拡張では考慮すべき課題でしょう。

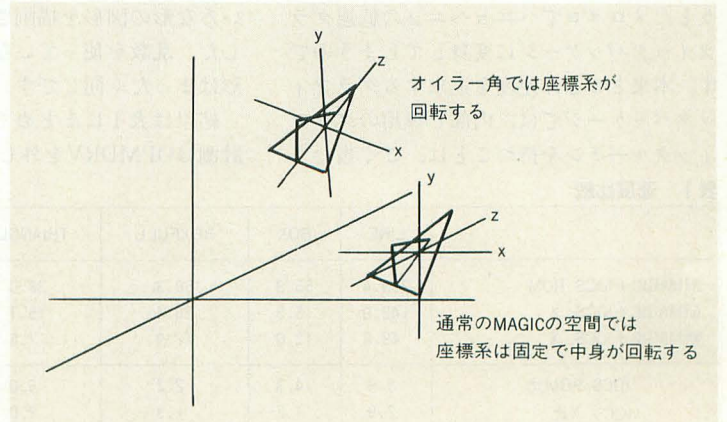
ポリゴン化の問題

ひととおりの高速化を終えたMAGICの次のターゲットはポリゴン化ということになるでしょう。現時点でもポリゴンは動いています。問題となるのは描画速度だけです。

こうなると画面解像度を下げるのは当然でしょう。512×512を256×256にするだけで4倍の高速化が可能です。X68000のG-RAMは色数によって速度が変わることはありませんから，存分に多色表示できます。

モノが画面全体の塗り潰しとなると，テキスト画面で行うことも考慮すべきでしょう。画面クリアにラスタコピーが使用できますし，水平方向の描画部分が桁違いに速い（256ドットならmovem命令ひとつですむ）のですから。ポリゴン表示の際はいかにラスタ描画を速くするかが大きな問題になるのです。

さて，単に塗り潰すだけでは3Dポリゴン表示はできません。ワイヤーフレームのときとは違って陰面処理が必要になってきます。

これは3D処理の本質的な問題に関わってきますが，リアルタイムゲームに使用するだけなら画面の奥のほうから描いていく「Zソート」あるいは，「リストプライオリティ」と呼ばれる方式で十分です。たまに不自然になりますが，本場のフライトシミュレータを見ても計算が常に正しいとは限りません。

3DCGはいかに正確に表示するかを中心にアルゴリズムを発展させてきましたが，ここでは多くのアルゴリズムがなしえたことを故意に無視していかなければなりません。不都合は「プログラムやデータ側で対処する」のが基本となります。そういえばSIONが作成されたときにはZ軸方向のクリッピングすらなかったのです。人間の都合にあわせた処理系を作ることよりも処理系の都合にあわせる技術を得ることのほうが有益となります。

いろいろ技を駆使すれば，けっこういいセンまでいきそうな気がするのですが……。

でもワイヤーが好き

M.N.M.の新作，開発コード「3D 2」はかなりいいデキです（ちなみにマジカルショットは3D 1）。日頃「3Dならここまでやらなきゃね」と思っていたことがほとんど盛り込まれていました。MAGICユーザーにもよい刺激となるでしょう。

オーガスタをやっていて再生モードがほしいと思った人はいませんか？ もっと自由に視点を変えられてもいいのでは……。

システムに関する認識度にもよりますが，せっかく作った3Dシステムなら，活用しないともったいないというものでしょう。同じデータ量のままで，さまざまな画面が作り出せるのですから。加えてワイヤーフレームの高速性は，ゲームの応答性のよさのみならず，存分に「見せる」演出ができるゲーム構成に貢献しています。

ポリゴンの美しさというのも確かにありますが，ワイヤーの美しさというのは「動き」に深く根差しています。私は思う存分「動き」を楽しめるワイヤーフレームが好きです。

図1 MAGICの空間

図2

ついに登場！　MAGIC高速版

グラフィックパッケージMAGIC ver.2.0

Kageyama Hiroaki 影山 裕昭

1991年5月号の付録ディスクに収録されたリアルタイムグラフィックパッケージMAGICがバージョンアップされました。IOCSを使わず，読者の皆さんに募集した高速ルーチンを厳選し，拡張モードを加えて……高速ver.2.0の誕生です。

高解像度，多色表示，高機能と三拍子揃ったX68000のグラフィック。グロテスクなグラフィックや画面いっぱいのデカキャラが登場するゲームは少なくありません。そんなゲームが氾濫しているからなのでしょうか。5月号の付録ディスクに収録したMAGICとサンプルゲームSIONは，流行のポリゴン表示を使わずに，ワイヤフレーム表示というまでは時代遅れの画面表示のゲームながら「よかった」といってくれた読者の方が実に多かったことに安心しました。

もちろん，山田氏と西川氏の努力があって，あそこまで素晴らしいゲームになったのでしょうが，MAGICを使うことによって，面倒な3次元から2次元への変換プログラムなどを考えることなく，ワイヤフレーム表示を実現していたことも事実です。もし，MAGICを使わずに同じゲームを作ろうとしたら，数倍の開発期間がかかったことでしょう。

しかし，MAGICとて問題がなかったわけではありません。なんといっても心苦しかったことは，グラフィックルーチンをすべてIOCSコールに頼っていたことです。付録ディスクを解凍したものはIOCS.Xを組み込むようになっていたので，そこそこの描画速度を実現していましたが，もしもIOCSコールを組み込むことを忘れてしまうと，メロメロでヘニョヘニョの低速グラフィックパッケージに変身してしまうのです。本来とことん速度を追求するグラフィックパッケージでは，内部で専用のグラフィックルーチンを持つことは，ごく当たり前のことです。それを怠っていた前バージョンは，「手抜き」と批判を受けても反論する余地のないところです。

しかし，こうしたことに理由がなかったわけではありません。開発期間が短かったこともありましたが，当時の僕がX68000のグラフィックまわりに関してまったくの素人だったということが最大の原因です（情けない話だが)。これではプログラムの組みようがないというものですし，僕が意地になって頑張ってプログラムを組むよりも，全国10万人を超えるX68000ユーザーの力を借りることが賢明だと思ったのです。結果として読者からも投稿があり，他力本願ながら転んでもただでは起きないぞ，といった形になりました。

新バージョンでは一部画面モードの設定にIOCSコールを使うほかは，グラフィックコマンドをすべてMAGIC内部のルーチンで処理しています。そのため前バージョンにIOCS.Xを組み込んだものより数倍速い処理速度を実現します。

速度を見てみる

実際の速さをわかってもらうには，言葉で説明するよりも数字で示したほうがいいだろうということで，簡単なテストをやってみました。テストはループを回していろいろな形の図形を描画させるものを使いました。乱数を使っていないので描画する図形はまったく同じです。

結果は表1にまとめておきました。なお，計測はOPMDRVを外した状態（10MHz）で行いましたが，手元のストップウォッチを使うという手抜きをしたので，±0.5秒ほどの誤差があるものと思ってください。また実行時間にはループや画面モードの設定など，グラフィック描画以外にかかる実行時間も含まれています。

表からIOCS.Xをはずした状態で前バージョンを動かすと，とんでもなく遅いことがよくわかるでしょう。しかし，IOCS.Xを組み込むと，ボックスだけはほかのコマンドと比べてIOCS.X版と新バージョンとの差が少ないことに目を引かれます。ところがIOCS.X V1.10のボックスは，マイナス値の座標を指定すると表示がおかしいというバグがあるようです。バグといえば大きな声ではいえないのですが，前バージョンにはライン，サークルに小さいバグがありました。今回，わかっているバグは排除してあるので安心して使ってください。

表1　速度比較

	LINE	BOX	BOXFULL	TRIANGLE	CIRCLE
旧MAGIC＋IOCS ROM	289.4	55.9	50.3	38.0	122.0
旧MAGIC＋IOCS. X	142.0	15.3	29.4	15.7	47.4
新MAGIC＋IOCS. X	49.0	13.0	22.9	7.6	18.9
IOCS ROM比	5.9	4.3	2.2	5.0	6.5
IOCS. X比	2.9	1.2	1.3	2.0	2.6

投稿プログラムについて

投稿プログラムは水平線を引くサブルーチンであるraster.sが，とっつきやすいこともあってか一番人気でしたが，そのなかにあって東京都の石黒育英さんがワイヤフレーム表示を受け持つdisp_flame.sを改良したものを送ってくれました。石黒さんの高速化のアイデアは，自作のラインルーチンを使ってグラフィックRAMにカラーコードを書き込むと同時に，バッファにドットを打つアドレスを保存しておき，裏画面消去はこのバッファからアドレスを取り出して，そこに0（これでドットを消す）を書き込むようにして行うというものでした。

つまり，消去時にはラインルーチンを使わずに，バッファからアドレスを取り出しては，0を書き込めばいいので，ループ展開を使って高速に画面を消去することができるのです。

しかし，

・バッファを大量に必要とする

・ラインルーチンの使い回しがきかない
といった悪い点もあります。

512×512の表示画面では262144点ものドットがあるわけです。いくらなんでもすべての点についてバッファを持つのはナンセンスですから，経験的観測に基づいてバッファの大きさを決めることになります。万が一にもバッファ容量を超えるドットを打つことになると，確実におかしくなります。システムに対する不信感が生まれるということで，これが欠点1。

さらにはラインルーチン内部でバッファにアドレスを格納するようになっているので，disp_flame.s以外でラインルーチンを使おうというときは余分な処理が入って速度が低下してしまうといったことがあります。これを嫌うなら別にラインルーチンを用意しなくてはいけません。

通常のライン描画が遅くなることと，アドレス計算がなくなることを天秤にかけてみます。ラインルーチンが速くなればなるほどこの方式のメリットは少なくなりそうです。

以上のことを考えた結果，メリットよりもデメリットのほうが大きいと判断し，今回は採用を見送ることにしました。やはりエラーの恐怖を背中に感じながらプログラミングするのは，精神衛生上よくないに決まってますからね。しかし，基本的な考え方については捨てがたいものもあるので，今後の動向次第ではこの方式を採用する可能性もあります。

次にraster.sは，投稿されてきたものをMAGICにインストールしてトライアングルの描画速度を計測し，最速のものを選出しました。その結果，愛知県の加賀和孝さんのプログラムを採用することにしました。

アルゴリズムは基本的にはどれも同じような処理でした。広いエリアをデータで埋める場合はmovemを使うのが高速ですので予想どおりといったところでしょうか。

ほかにも，ライン，ボックスフル，サークルフルについても加賀さんの投稿してくれたプログラムを採用しています。アルゴリズムの解説がレポート用紙に詳しく書かれていたこともよかったですね。おや，加賀さんは受験生じゃないですか。マシンの封印は考えていないということだけど，大学受験も頑張ってくださいね。ところで投稿されてきたものは，まだまだ速いコードに書き換えられる部分があったので，掲載版は加賀さんのプログラムに若干の変更を加えたものを使用しています。ご了承ください。

入力方法

以前に将来のコマンドの改良・拡張の話に触れて，モジュールごとに分割して開発する利点について話したと思います。コマンドの改良を行うには，対応するモジュールを変更することになるのですが，新バージョンではほとんどすべてのモジュールに手が加えられ，結局para.s以外すべてのモジュールが変更されています。

ダンプリストから入力する場合は68ページを見てください。

ソースリストから入力する場合はリスト1からリスト21をすべて打ち込んだら，付録ディスクに収録されていたpara.sを加えてリスト22のバッチファイルを実行してください。もしもエラーがあったらリストを見直して間違いを訂正して再度実行してください。

モジュール解説

これから各モジュールを解説します。プログラムは前バージョンのものを参考にしていますが，メモリにワークを取る代わりに空いているレジスタ（アドレスレジスタも）を使っているので，解析は相当に難しいかもしれません。プログラムを高速化していくと，どうしても第3者が読みづらいプログラムになってしまいます。改良点，変更点については該当モジュールで詳解しますが，大きな改良点を述べておくことにしましょう。

・数倍の処理速度向上
・ワイヤフレーム表示におけるZ軸のクリッピング
・画面モードを追加
・物体ごとに描画色が設定可能（拡張モード）
・ポイントコマンドを削除

それでは見ていくことにしましょう。

●ライン　line.s

ラインルーチンはMAGICのなかでももっとも高速化が待ち望まれていたものです。特にワイヤフレーム表示ではラインルーチンの速度によって，処理速度がまるで違うといっても過言でないでしょう。

新バージョンはラインルーチンに読者投稿のものを使用しています。加賀さんは6年前，MZ-700上でオールBASICのピコピコシューティングを作っていて，敵弾移動ルーチンに「X座標とY座標それぞれの差分を取って大きいほうで小さいほうを割り，商を増分として足していく」方法を思いついたのだそうです。

投稿されてきたラインルーチンは，そのアルゴリズムを使っているようですが，8倍展開＋両端から同時書きのテクニックも使いさらに高速描画を実現していました。結果的に割り算を使っているにも関わらず，gline（マシン語プログラミング掲載のラインルーチン）より速くなっています。6月号に掲載されたラインルーチンの処理速度を測るプログラムに，このラインルーチンを組み込んだ結果を示します（OPMDRVを外した状態。10MHzでの測定結果）。

gline	41.08秒
加賀line	33.94秒

クリッピングルーチンは掛け算，割り算を使う単純なものです。加賀さんは5月号の村田さんの記事を読んで，本当に速いものかと思い，実際にクリップするところのみクロック数を数えてみたのだそうです（大変だったでしょうね)。結果は加賀さんのルーチンでdivs，mulsに最悪かかったとしても，村田さんのルーチンが3回で収束するのと同じクロック数だったそうです。3回以内で収束する場合は少ないだろうということで，このクリッピングルーチンを使ったそうで，完全可視の場合の負担が軽いように作られていました（リスト2）。

投稿版：102または112クロック
gline　：498クロック
その差396クロック×50,000本＝1.98秒

それから投稿プログラムをこちらで描画モードに対応させました。描画モードごとにルーチンを用意してもよかったのですが，ここでは自己書き換えを使ってVRAMへの書き込み方法を変更しています。MODEコマンドで設定した描画モードが有効になります。なお，画面左上のVRAMのアドレスを，ラベルl_gbase+2に格納することで書き込むグラフィックページを指定します。

前バージョンではグラフィック描画にIOCSコールを使ったので，書き込みページの指定もIOCSコールを使っていましたが，新バージョンではグラフィックルーチンがIOCSから独立したので，書き込みページを切り替えるAPAGEコマンドをMAGICに追加しました。詳しくはあとで解説します。

また，ラベルl_pixel,l_sft_timeは実画面サイズに応じてscrmod.sによって書き換えられます。似たようなラベルはbox,circle,boxfull,rasterにもありますが，すべて同じ意味を持ちます。

なお，line_mainのラベルを定義していま

すが，これは(d1.w,d2.w)—(d3.w,d4.w)のラインを引くサブルーチンで破壊レジスタはd1～d7，a1，a2です。

●スプライン　spline.s

このプログラムは，はっきりいって僕にはなにをやっているのかわからないので，Z80からの移植のままで終わっています。ソースリストを見ると曲線を区間に分割して区間ごとに線分を求める，といったことはわかるのですが，具体的な計算式がまったくわからずお手上げ状態です。高速化のための変更点はワークエリアを使わずに，余っているレジスタを使うようにしたことくらいです。

●ボックス　box.s

最初はラインルーチンを使ってボックスを描いて終わりにしようと思っていたのですが，それだとIOCS.Xのほうが速いことがわかり，結局クリッピングとアドレス計算をしたあと，横ラインはrasterを呼んで描画し，縦ラインは8倍展開したループを回しています。それにしてもROMのIOCSは遅いですね。

●トライアングル　triangle.s

アルゴリズム（スキャンラインコンバージョン）は6月号に掲載したプログラムと変わりありませんが，ループの中でメモリに持たせていた情報を空いていたレジスタに持たせることによって高速化を図っています。

プログラムの中で自分自身を書き換えている部分がありますが，これはxを増やすのか，減らすのか，不変なのか，といった情報を3点の座標値によって変える必要があるからです。これは自己書き換えといって，メモリ容量の節約と処理の高速化に役立つ方法です。テクニックとはとうてい呼べるものではなく，低レベルで68000には似合わない手法だと思いますが，皆さんの意見はどうでしょうか。このモジュールは2次元グラフィックコマンドのなかで，高速化に一番力を入れました。なぜだかわかるかな？

●ボックスフル　boxfull.s

このプログラムは加賀さんによる投稿です。送られてきたものは縦，横のドット数を(x2−x1)，(y2−y1)で計算していましたが，実際はこれに1を足す処理が必要です。これが抜けていたので，僕が直しておきました。プログラムはraste_rmainを何回も呼んでいるだけです。これはIOCSも結構速いので，差はあまりついていませんね。

●サークル　circle.s

これも加賀さんによるプログラムです。原稿によると，元のプログラムが相当に見にくかったので（こりゃ失礼），Z80's Barの説明を見ながら初めから作り直したそうです。説明をした（で）氏もこれを聞いたら喜ぶことでしょう。クリッピング不要のときはraster_mainを，必要なときはrasterを呼ぶように区別して高速化を図っています。また，前バージョンにあった非常に小さくなるとおかしくなる，および，上下の端に1ドット出ていてみっともない，というバグも取られています。感謝！

●ウィンドウ　window.s

前バージョンではIOCSコールWINDOWを呼んでいましたが，新バージョンではMAGIC内部のワークエリアにデータ転送するだけです。またウィンドウ設定にともなって，透視変換の際に必要となるウィンドウの中心座標をラベルpoint_ox1＋2，point_ox2＋2，point_oy1＋2，point_oy2＋2に転送します。

●モード　mode.s

描画モード（0～3）をラベルdisp_modeに転送したあと，モードに応じてラインルーチン，ラスタルーチンを書き換えます。a1.l＝グラフィックRAMのアドレス，d0.w＝カラーコードとすると，1ドットの書き込みは描画モードによって次のように書き換えられます。

```
preset    clr.w  (a1)
xor       eor.w  d0,(a1)
or        move.w d0,(a1)
nop       nop
```

●ポイント

このコマンドの必要性があるのか周りの人の意見を聞いたうえで，必要ないと判断し削除しました。したがってこのコマンドを呼び出してもなにもせずに復帰します。同等のコマンドがIOCSコールPOINTにありますから，そちらを利用してください。

●クリアスクリーン　wipe.s

最初はIOCSコールWIPEを使っていたものをDMAを使ってグラフィックRAMをクリアするよう変更したのですが，ウエイトが入ってしまうせいか思ったほど速くなかったので没にしました。というわけで，MAGICのボックスフルで黒で描画することで画面クリアしています。

●セット3Dパラメータ　para.s

唯一変更のなかったモジュールです。透視変換で使うパラメータをワークに転送しているだけです。

●セット3Dデータ　data.s

物体の頂点データと線分データをワークに設定するコマンドですが，線分データの格納方法を変更しました。また後述する拡張モードでは，データフォーマットに描画色の指定が加わります（前バージョンとのデータ互換性がなくなってしまうが）。

前バージョンのMAGICは線分データをバッファ（lin_buf）に格納していくときに，以前に登録した物体の線分数をオフセットとして加えています。複数（同一）の物体を2回以上データセットする場合を考えてください。登録した物体はその都度透視変換をしてバッファ(disp_buf)に格納されていきます。このときに頂点番号にオフセットを加えないと，物体1の頂点0なのか，物体2の頂点0なのか，はたまた物体3の頂点0なのか区別がつかなくなってしまいます。そこで直前に登録した物体が線分4本で構成されていたら，次の物体はすべての頂点番号にオフセット（この場合は4ですが）を加えるようにして内部で区別していました。

新バージョンではデータ構造を変更して，ひとつの物体を8バイトの構造体で表すようにしました（図1）。この構造体はdata_bufに200個分用意されています。すなわち登録できる物体の最大数は200になります。200を超えるとエラーが発生するでしょうから注意してください。

本来ならば線分データの格納アドレスもポインタで与えるようにして，構造体に加えてやればデータ転送がなくなり高速化が実現できるのですが，SIONのアニメーションがバッファ内のデータを書き換えて行うようになっているため，やむを得ずこのままのかたちとしました。今後の動向によっては変更もありえます。

データ転送は16倍にループ展開したものを使い，登録した物体数をラベルobj_noに格納します。

●3D－2D変換　perspective.s

乗除算の見直しと，ループの中で空いていたレジスタにデータをセットすることによって高速化を図っています。このモジュールは（x，y，z）を3Dパラメータに従ってアフィン変換をして（x′，y′，z′）を求め，さらにそれを透視変換して2D変換された（x″，y″）を求めます。

またワイヤフレーム表示でZ軸をクリッ

図1　物体構造体

```
+0L  2D変換後の座標格納アドレス
+4W  物体を構成する線分の数
+6W  描画色（拡張モードで使用）
     L=LONG WORD
     W=WORD
```

ピングするためにアフィン変換後の（x′, y′, z′）が必要なので，透視変換後の座標データに続けて格納します(図2)。ひとつの座標は16バイトの領域を必要とし，バッファ(disp_buf)に頂点の数だけ繰り返し格納していきます。

●ワイヤフレーム表示　disp_flame.s

lin_bufとdisp_bufに格納された情報からワイヤフレーム表示をします。グラフィックページが2枚以上ある場合は，画面切り替えを使って画面のちらつきをださないようにしています。

ここでのページ切り替えはAPAGEを呼ばずに，ラインとラスタの書き込みページだけを直接書き換えています。というのはAPAGEは関係のないサークルやボックスの書き込みページまで変更するので，処理速度が余計にかかるのを嫌ったからです。そのため，このコマンド終了後は書き込みページがグラフィックルーチンによってバラバラになっている場合がありますので，ラインやスプラインを使うときはAPAGEで書き込みページを再指定するようにしてください。

なお，新たに画面モードにグラフィック4面モードが設けられましたが，ワイヤフレーム表示はページ0，1をシステムで使用していますので，背景などはページ2，3を使用して描いてください。

裏画面の消去は前バージョンと同じく，同じ物体を黒で再描画することでやっています。ですからMODEコマンドで描画モードをxor, nopに指定すると，裏画面が消去されずに残ってしまいますので注意してください。

また，新バージョンではZ軸のクリッピングをするようにしました。と聞いてもピンとこない人がいるかもしれませんから説明しましょう。車に乗っているときのことを考えてみてください。ドライバーは常に前方に視線を集中させます。目に入る信号機は最初小さく見えても，近づくにつれて大きくなり，ついには視界の後方にいってしまいます。バックミラーを使うか，さもなくば後ろを振り返らなければ見ることができません。また，限りなく長い直線があると仮定して，前を走っている車が自分より速ければだんだんと小さくなり最後は見えなくなってしまうでしょう。

このように視界の後ろにいった物体は見えてはいけないし，遠いところにある物体を見るときも，個人差はあるにせよ人間が物体を認識することのできる距離には限界があります。この基本的な自然現象を前バージョンのMAGICではシミュレートしていませんでした（バグではなくシミュレートしてなかったのだ）。

付録ディスクに収録したtyrrel.xを実行して，Zキーを押してみてください。物体はどんどん手前に近づいてきて頭上を越えていくと思いきや，スクリーンに張りついたままいつまでも表示されています。今度はAキーを押してみます。物体は遠ざかっていきますが，いつまでたってもスクリーンから消えません。これがZ軸についてクリッピングすると，近づいてきた物体はそのまま頭上を越えて見えなくなり，遠くにある物体は見えなくなってしまいます。なおZ軸のクリッピングは，$z<0$，$z>maxz$で中点分割アルゴリズムを使ってやっています。なおmaxzは新たに設けられたdepthコマンドでプログラマが自由に設定することができます。

●描画色指定　set_color.s

描画色を変更します。説明はいらないでしょう。

●画面モード指定　scrmod.s

新たに16色4画面モードを使えるようにしました。いま考えてみると，なぜ最初からIOCSコールコンパチに画面モードを用意しなかったのか自分でも不思議です。

新バージョンではIOCSコールコンパチにしようとも思いましたが，前バージョンのMAGICとの互換性を考えてこのようにしました。画面モードを変更すると，同時に表示画面の大きさにウィンドウが設定され（透視変換に必要な情報も書き込まれます）ますので，表示画面いっぱいにグラフィック表示する場合は，わざわざウィンドウを再設定する必要はありません。

ここでは指定した画面モードが実画面1024×1024の場合と512×512の場合で，内部のグラフィックルーチンを書き換えています。これは実画面の大きさによってグラフィックRAMのアドレス計算に必要な定数の値が違ってくるからです。ですから，画面モードの変更は必ずこのコマンドを使ってください。

またモード番号に＄100を足して(つまり第8ビットを1にして）コマンドを呼び出すと，拡張モードを指定したことになります。拡張モードでは物体ごとに描画色を指定してワイヤフレーム表示することができます。

●初期化　init.s

ワークエリアをDMA転送を使って初期化します。そのほかのワークに初期値を設定します。

●登録，呼び出し　magic.s

コマンド番号を解析して，対応するサブルーチンを呼び出します。ほとんど変更はありませんが，コマンドの削除，追加にともなってジャンプテーブルを変更しています。またテーブルから呼び出すサブルーチンのアドレスを計算するところを最適化してあります。これによってすべてのコマンドの呼び出しが速くなりました。

●ラスタ　raster.s

加賀さんは，元は1024倍展開という恐ろしいことをやっていたらしいのですが，IOCSと大差ないことに気づき，しばらく悩んだ末思いついたのがmovem.lによる16ドット一気書き込み＋64倍展開の方法だそうです。たぶんこれより速くするには，DMAを使うしかないでしょう。

raster_mainのラベルは，a1はラスタの右端のアドレスを入れ，d1に長さを入れてコールするとラスタを描きます。いちいちアドレスを計算する手間を省くため，circle.s, boxfull.sで使用するために作りました。d0～d2，a1，a2レジスタを破壊します。

●sin，cos値を得る　sincos.s

説明するまでもなくテーブルから値を取り込むだけです。前バージョンではd1.wに値を格納しましたが，新バージョンは取り込まれた値が，move.w d1,(a1)＋で(a1)に格納され，a1の値が1ワードだけ増加します。

●書き込みページ指定　apage.s

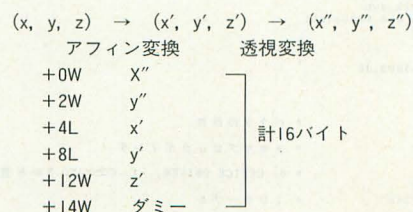
ワイヤフレーム表示以外のグラフィックコマンドの書き込みページを指定します。ラベルapage_mainはd1.wにページ番号（0～3）を入れて呼び出します。

●投影面，後方面の設定　depth.s

ワイヤフレーム表示の投影面と後方面を設定するコマンドで，ラベルminxに投影面のZ値を，ラベルmaxzに後方面のZ値を格納します。投影面のZ値を大きくすると画面奥（？）に投影面が置かれ，小さくす

図2

(x, y, z) → (x′, y′, z′) → (x″, y″, z″)
アフィン変換　透視変換

+0W	X″	計16バイト
+2W	y″	
+4L	x′	
+8L	y′	
+12W	z′	
+14W	ダミー	

ると画面手前に投影面が置かれます。後方面も効果は同様ですが，後方面より大きな線分データはz＝maxzでクリッピングします。後方面を指定すると遠くの物体を表示しないようにすることが簡単にできるので，フライトシミュレータなどを作るときに役に立つかもしれません。ちなみに初期値は投影面が50，後方面が2000です。

拡張モードについて

拡張モードの指定はモジュール解説でも書いたように，SCRMODで指定する画面番号に$100 (256) を足して行います。拡張モードでは３Ｄ物体をデータセットするときに描画色の指定のために１ワードデータを加える必要があります。このデータを設定しないと暴走するので注意してください。具体的には線分数の後ろに描画色を指定します。表２にあるコマンド一覧表を参照してください。

Ver2.00の開発を終わって

MAGICはパラメータに規定外の数値を与えても，エラーメッセージを出して実行を停止することはありません。データ管理をプログラマに任すことによって，データチェックを省き処理の高速化を目指しているからです。DOSコールやIOCSコールを使いなれている人たちから見れば，非常に扱いにくいと感じるかもしれません。しかし使い方さえ誤らなければ，MAGICはあなたの右腕として非常に大きな力を発揮してくれることでしょう。僕はひとりでも多くの方にMAGICを使っていただきたいと思っています。

まだまだ高速化の余地は残されている

付録ディスクのように目で速度を見てもらうことができないのが残念ですが，このバージョンでMAGICもかなり使えるレベルに達したのではないかと思います。しかし，まだまだ改良できる点はあります。現在のモジュールをこれ以上最適化することは難しいですから，やるならアルゴリズムの変更になるでしょう。ちょっと考えてみても，データセットの転送をやめることや，パラメータ設定を複数まとめて行うようにするとかいった方法が考えられます。データの指定法などの改善の余地があります。

MAGICは立ち止まることを知りません。今後も細かいバージョンアップが続けられることになるでしょう。

最後にあまり大きな声ではいえないのですが，僕の手元ではすでにポリゴン対応のMAGICが動いています。とりあえずポリゴン表示ができるといった，まだまだ評価版といったデキですが，もしかしたら次のディスクにはこの評価版，ひょっとしたらV1.00になったものが収録できるかもしれません。ご期待ください。

リスト１

```
  1: ************************************************************
  2: * 高速グラフィックパッケージ ＭＡＧＩＣ　ＴＵＸ吉村
  3: *
  4: * Ｚ軸クリッピングバージョン
  5: *
  6: *                                   converted by SHADOW
  7: ************************************************************
  8:
  9:         .include        doscall.mac
 10:         .include        iocscall.mac
 11:
 12:         .xref   line
 13:         .xref   spline
 14:         .xref   box
 15:         .xref   triangle
 16:         .xref   boxfull
 17:         .xref   circle
 18:         .xref   window
 19:         .xref   mode
 20:         .xref   cls
 21:         .xref   para
 22:         .xref   data
 23:         .xref   perspective
 24:         .xref   disp_flame
 25:         .xref   scrmod
 26:         .xref   set_color
 27:         .xref   init
 28:         .xref   apage
 29:         .xref   depth
 30:         .xref   p_end
 31:
 32:         .text
 33:         .even
 34:
 35: device_header:
 36:         dc.l    -1
 37:         dc.w    $8000
 38:         dc.l    strategy_entry
 39:         dc.l    interrupt_entry
 40:         dc.b    'MAGIC***'
 41: request_header:
 42:         dc.l    0
 43: strategy_entry:
 44:         move.l  a5,request_header
 45:         rts
 46:
 47: interrupt_entry:
 48:         movem.l d0/a5,-(sp)
 49:         movea.l request_header,a5
 50:         clr.l   d0
 51:         move.b  $0002(a5),d0
 52:         bsr     initialize
 53:         move.b  d0,$0003(a5)
 54:         lsr.w   #8,d0
 55:         move.b  d0,$0004(a5)
 56:         movem.l (sp)+,d0/a5
 57:         rts
 58:
 59: initialize:
 60:         bne     init_sub
 61:         movem.l d1/a1,-(sp)
 62:         clr.w   device_flg
 63:         bsr     magic_intvcs
 64:         movem.l (sp)+,d1/a1
 65:         beq     init_sub
 66:         move.l  #p_end,$000e(a5)
 67:         clr.l   d0
 68:         rts
 69: init_sub:
 70:         move.w  #$5003,d0
 71:         rts
 72:
 73: save_vecter:
 74:         ds.l    1                       * ベクタの保存
 75: memptr:
 76:         ds.l    1                       * メモリブロックポインタ
 77: device_flg:
 78:         dc.w    0                       * 0..DEVICE DRIVER, -1..COMMAND.Xから登録
 79: id_table:
 80:         dc.l    'FDfn'                  * ＩＤテーブル
 81: id_table2:
 82:         dc.l    'FEfn'
 83: fdfn_vecter:
 84:         movem.l d6/a5-a6,-(sp)
 85:         lea.l   $000e(sp),a6
 86:         movea.l (a6),a5
 87:         move.w  (a5)+,d6
 88:         sub.w   #$fd00,d6
 89:         cmp.w   #$100,d6
 90:         bcs     fd_function     * $FD00≦d6≦$FDFF
 91:         movem.l (sp)+,d6/a5-a6
 92:         move.l  save_vecter,-(sp)
 93:         rts
 94:
 95: fd_function:
 96:         move.l  a5,(a6)+
 97:         btst.b  #$05,$000c(sp)
 98:         bne     fd_func2
 99:         move.l  usp,a6
100: fd_func2:
101:         movem.l d0-d5/d7/a1-a4,-(sp)
102:         add.w   d6,d6
103:         move.w  jmptbl(pc,d6.w),d6
104:         jsr     jmptbl(pc,d6.w)
105:         movem.l (sp)+,d0-d5/d7/a1-a4
106:         move.w  sr,d6
107:         move.b  d6,$000d(sp)
108:         movem.l (sp)+,d6/a5-a6
109:         btst.b  #$07,(sp)
110:         bne     fd_func3
111:         rte
112: fd_func3:
113:         ori.w   #$8000,sr
114:         rte
115: auto:
116:         move.w  (a0)+,d0
117:         add.w   d0,d0
118:         pea.l   auto(pc)
119:         move.w  jmptbl(pc,d0.w),d0
120:         jmp     jmptbl(pc,d0.w)
121: jmptbl:
122:         dc.w    line-jmptbl                     * fd00
123:         dc.w    spline-jmptbl                   *    1
124:         dc.w    box-jmptbl                      *    2
125:         dc.w    triangle-jmptbl                 *    3
126:         dc.w    boxfull-jmptbl                  *    4
127:         dc.w    circle-jmptbl                   *    5
128:         dc.w    window-jmptbl                   *    6
129:         dc.w    mode-jmptbl                     *    7
130:         dc.w    no-jmptbl                       *    8
131:         dc.w    cls-jmptbl                      *    9
132:         dc.w    no-jmptbl                       *    a
133:         dc.w    para-jmptbl                     *    b
134:         dc.w    data-jmptbl                     *    c
135:         dc.w    perspective-jmptbl              *    d
136:         dc.w    disp_flame-jmptbl               *    e
137:         dc.w    done-jmptbl                     *    f
138:
139:         dc.w    set_color-jmptbl                *   10
140:         dc.w    scrmod-jmptbl                   *   11
141:         dc.w    init-jmptbl                     *   12
142:         dc.w    auto-jmptbl                     *   13
143:         dc.w    apage-jmptbl                    *   14
144:         dc.w    depth-jmptbl                    *   15
145: magic_intvcs:
146:         move.l  #$0b,d1
147:         lea.l   fdfn_vecter(pc),a1
148:         IOCS    _B_INTVCS       * F line エミュレータのベクタを変更する
149:         move.l  d0,save_vecter  * 現在のベクタを保存
150:         movea.l d0,a1
151:         suba.l  #4,a1
152:         IOCS    _B_LPEEK
153:         cmp.l   id_table2,d0    * 'FEfn'と比較
154:         beq     magic_intvcs1
155:         clr.l   id_table2
156: magic_intvcs1:
157:         subq.l  #8,a1
158:         move.l  a1,a3           * A1.LをA3.Lに退避
159:         IOCS    _B_LPEEK
160:         cmp.l   id_table,d0     * 'FDfn'と比較
161:         bne     magic_message   * ＭＡＧＩＣが登録されていないので登録する
162:
163:         movea.l save_vecter,a1
164:         moveq.l #$0b,d1
```

```
165:         IOCS    _B_INTVCS           * ベクタを元に戻す
166:
167:         tst.b   (a2)+
168:         beq     exact               * パラメータがない
169:
170:         bsr     skip_chr            * タブとスペースをスキップする
171:         bsr     param_chk           * パラメータチェック
172:         tst.w   d7
173:         bmi     magic_intvcs3
174:
175:         move.l  a3,a1               * 退避したA1.L
176:         subq.l  #2,a1
177:         IOCS    _B_WPEEK
178:         tst.w   d0
179:         bne     magic_intvcs2       * COMMAND.Xから組み込まれた
180:
181:         pea.l   magic_no
182:         DOS     _PRINT              * システムに組み込まれている
183:         addq.l  #4,sp
184:         bra     magic_intvcs3
185: magic_intvcs2:
186:         pea.l   magic_kaijo
187:         DOS     _PRINT
188:         addq.l  #4,sp
189:
190:         subq.l  #6,a3
191:         move.l  (a3),-(sp)
192:         DOS     _MFREE              * メモリを開放する
193:         addq.l  #4,sp
194:         subq.l  #4,a3
195:         move.l  (a3),a1
196:         IOCS    _B_INTVCS           * ベクタを元に戻す
197: magic_intvcs3:
198:         clr.l   d0
199:         rts
200:
201: usage:
202:         pea.l   usage_mes
203:         DOS     _PRINT              * 使い方を表示
204:         addq.l  #4,sp
205:         moveq   #$ff,d7
206:         bra     magic_intvcs3
207: exact:
208:         pea.l   exact_mes
209:         DOS     _PRINT              * 使い方を表示
210:         addq.l  #4,sp
211:         bra     magic_intvcs3
212:
213: magic_message:
214:         tst.w   device_flg
215:         beq     message2
216:         tst.b   (a2)+
217:         beq     message2
218:         bsr     skip_chr            * タブとスペースをスキップする
219:         bsr     param_chk           * パラメータチェック
220:
221:         movea.l save_vecter,a1
222:         moveq.l #$0b,d1
223:         IOCS    _B_INTVCS           * ベクタを元に戻す
224:
225:         tst.w   d7
226:         bmi     magic_intvcs3
227:
228:         pea.l   no_exact
229:         DOS     _PRINT
230:         addq.l  #4,sp
231:         bra     magic_intvcs3
232: message2:
233:         pea.l   mes1
234:         DOS     _PRINT              * ＭＡＧＩＣ登録メッセージ
235:         addq.l  #4,sp
236:         moveq.l #$ff,d0
237:         rts
```

```
238:
239: magic_com:
240:         move.w  #$ff,device_flg
241:         bsr     magic_intvcs
242:         beq     magic_end
243:         add.l   #$10,a0             * メモリブロックのポインタ
244:         move.l  a0,memptr
245:         clr.w   -(sp)
246:         move.l  #p_end-device_header,-(sp)
247:         DOS     _KEEPPR             * 常駐終了
248:
249: magic_end:
250:         DOS     _EXIT
251:
252: no:
253:         rts
254: done:
255:         addq.w  #4,sp    * ワードなのは間違いじゃありません
256:         rts
257:
258: skip_chr:
259:         move.b  (a2)+,d0
260:         cmpi.b  #8,d0               * タブコード
261:         beq     skip_chr
262:         cmpi.b  #' ',d0             * スペース
263:         beq     skip_chr
264:         rts
265:
266: param_chk:
267:         cmpi.b  #'/',d0
268:         beq     param_chk2
269:         cmpi.b  #'-',d0
270:         bne     usage
271: param_chk2:
272:         move.b  (a2),d0
273:         andi.b  #%1101_1111,d0  * 大文字にする
274:         cmpi.b  #'R',d0
275:         bne     usage
276:         moveq   #0,d7
277:         rts
278:
279:         .data
280:
281: usage_mes:
282:         dc.b    'High speed graphic package ＭＡＧＩＣ version 2.00',13,10
283:         dc.b    '使用法：magic [スイッチ]',13,10
284:         dc.b    '        /r      ＭＡＧＩＣの常駐解除',13,10,0
285:
286:         .even
287:
288: no_exact:
289:         dc.b    'ＭＡＧＩＣは登録されていません',13,10,0
290:         dc.b    0
291: exact_mes:
292:         dc.b    'ＭＡＧＩＣは既に組み込まれています',13,10,0
293:         dc.b    0
294: magic_kaijo:
295:         dc.b    'ＭＡＧＩＣの占有していたメモリを開放します',13,10,0
296:         dc.b    0
297: magic_no:
298:         dc.b    'ＭＡＧＩＣの占有しているメモリは開放できません',13,10,0
299:         dc.b    0
300: mes1:
301:         dc.b    13,10
302:         dc.b    '高速グラフィックパッケージ ＭＡＧＩＣ version 2.00',13,10
303:         dc.b    'Original TUX-Yoshimura SEP.1986 Oh!X',13,10
304:         dc.b    'Converted & improved by Shadow Mountain MAY.1991 Oh!X',13,10
305:         dc.b    'Improved by Shadow Mountain, KAZ SEP.1991 Oh!X',13,10
306:         dc.b    0
307:
308:         .end    magic_com
309:
```

リスト2

```
  1: *
  2: *  line.s
  3: *
  4:         .xdef   line
  5:         .xdef   line_main
  6:         .xdef   l_gbase
  7:         .xdef   l_pixel
  8:         .xdef   l_sft_time
  9:         .xdef   x1
 10:         .xdef   y1
 11:         .xdef   x2
 12:         .xdef   y2
 13:         .xdef   x3
 14:         .xdef   y3
 15:
 16:         .xdef   Xline_draw
 17:         .xdef   Yline_draw
 18:         .xdef   line45_draw
 19:         .xdef   dot2
 20:         .xdef   l_putch1
 21:         .xdef   l_putch2
 22:         .xdef   l_putch3
 23:         .xdef   l_putch4
 24:         .xdef   l_putch5
 25:         .xdef   l_putch6
 26:         .xdef   l_putch7
 27:
 28:         .xref   raster
 29:         .xref   color
 30:         .xref   minx
 31:         .xref   miny
 32:         .xref   maxx
 33:         .xref   maxy
 34:         .xref   disp_mode
 35:
 36:         .text
 37:         .even
 38:
 39: line:
 40:         move.w  (a0)+,d6
 41:
 42:         subq.w  #2,d6
 43: line_loop:
 44:         move.w  d6,-(sp)
 45:         movem.w (a0)+,d1-d2         * x0,y0
 46:         movem.w (a0),d3-d4          * x1,y1
 47:         jsr     line_main(pc)
 48: line_end:
 49:         move.w  (sp)+,d6
 50:         dbf     d6,line_loop
 51:         addq.w  #4,a0
 52: return:
 53:         rts
 54: clip_maxx:                          * クリッピング用サブルーチン
 55:                                     * 素直に書いてあります。
 56:         cmp.w   maxx(pc),d1
 57:         bgt.b   return
 58:         move.w  maxx(pc),d0
 59:         sub.w   d3,d0
 60:         muls    d6,d0
```

```
 61:         divs    d5,d0
 62:         add.w   d0,d4
 63:         move.w  maxx(pc),d3
 64:
 65:         move.w  d3,d5
 66:         sub.w   d1,d5
 67:         move.w  d4,d6
 68:         sub.w   d2,d6
 69:         bra     clip1
 70: clip_minx:
 71:         cmp.w   minx(pc),d3
 72:         blt.b   return
 73:         move.w  minx(pc),d0
 74:         sub.w   d1,d0
 75:         muls    d6,d0
 76:         divs    d5,d0
 77:         add.w   d0,d2
 78:         move.w  minx(pc),d1
 79:
 80:         move.w  d3,d5
 81:         sub.w   d1,d5
 82:         move.w  d4,d6
 83:         sub.w   d2,d6
 84:         bra     clip2
 85: clip_maxy1:
 86:         cmp.w   maxy(pc),d2
 87:         bgt.b   return
 88:         move.w  maxy(pc),d0
 89:         sub.w   d4,d0
 90:         muls    d5,d0
 91:         divs    d6,d0
 92:         add.w   d0,d3
 93:         move.w  maxy(pc),d4
 94:
 95:         move.w  d3,d5
 96:         sub.w   d1,d5
 97:         move.w  d4,d6
 98:         sub.w   d2,d6
 99:         bra     clip3
100: clip_miny1:
101:         cmp.w   miny(pc),d4
102:         blt     return
103:         move.w  miny(pc),d0
104:         sub.w   d2,d0
105:         muls    d5,d0
106:         divs    d6,d0
107:         add.w   d0,d1
108:         cmp.w   maxx(pc),d1
109:         bgt     return
110:         move.w  miny(pc),d2
111:
112:         move.w  d3,d5
113:         sub.w   d1,d5
114:         move.w  d4,d6
115:         sub.w   d2,d6
116:         bra     l_gbase
117: clip_maxy2:
118:         cmp.w   maxy(pc),d4
119:         bgt     return
120:         move.w  maxy(pc),d0
```

```
121:        sub.w   d2,d0
122:        muls    d5,d0
123:        divs    d6,d0
124:        add.w   d0,d1
125:        cmp.w   maxx(pc),d1
126:        bgt     return
127:        move.w  maxy(pc),d2
128:
129:        move.w  d3,d5
130:        sub.w   d1,d5
131:        move.w  d4,d6
132:        sub.w   d2,d6
133:        bra.b   clip4
134: clip_miny2:
135:        cmp.w   miny(pc),d2
136:        blt     return
137:        move.w  miny(pc),d0
138:        sub.w   d4,d0
139:        muls    d5,d0
140:        divs    d6,d0
141:        add.w   d0,d3
142:        move.w  miny(pc),d4
143:
144:        move.w  d3,d5
145:        sub.w   d1,d5
146:        move.w  d4,d6
147:        sub.w   d2,d6
148:        bra.b   line_13
149:
150: line_main:
151:        moveq.l #0,d5
152:        moveq.l #0,d6
153:
154:        move.w  d3,d5
155:        sub.w   d1,d5
156:        bpl.b   line_11 * x2>=x1
157:        exg     d1,d3
158:        exg     d2,d4
159:        neg.w   d5          * d5.w = dx(>=0)
160: line_11:
161:        move.w  d4,d6
162:        sub.w   d2,d6
163:        beq     call_raster        * 水平なら
164: l_pixel:
165:        move.w  #1024,d7           * 1ラインのドット数*2
166:                                   * 1024 or 2048
167:
168:        cmp.w   maxx(pc),d3                * Clipping
169:        bgt     clip_maxx
170: clip1: cmp.w   minx(pc),d1
171:        blt     clip_minx
172: clip2:
173:        tst.w   d6
174:        bmi.b   clip_y2
175: clip_y1:
176:        cmp.w   maxy(pc),d4
177:        bgt     clip_maxy1
178: clip3: cmp.w   miny(pc),d2
179:        blt     clip_miny1
180:        bra.b   l_gbase
181: clip_y2:
182:        cmp.w   maxy(pc),d2
183:        bgt     clip_maxy2
184: clip4: cmp.w   miny(pc),d4
185:        blt     clip_miny2
186: line_13:
187:        neg.w   d6                 * 符号反転
188:        neg.w   d7
189: l_gbase:
190:        lea.l   $c00000,a1         * vram top address
191:        movea.l a1,a2
192:        ext.l   d2
193:        ext.l   d4
194: l_sft_time:
195:        moveq.l #10,d0
196:
197:        lsl.l   d0,d2
198:        add.w   d1,d1
199:        add.w   d1,d2
200:        adda.l  d2,a1              * a1 = 左端の点
201:
202:        lsl.l   d0,d4
203:        add.w   d3,d3
204:        add.w   d3,d4
205:        adda.l  d4,a2              * a2 = 右端の点
206:
207:        move.w  color(pc),d0
208:
209:        move.w  #$8000,d1          * 点をドットのど真ん中にする。
210:
211:        cmp.w   d5,d6              * dY-dX
212:        beq     LINE45
213:        bpl     YLINE
214: XLINE:
215:        swap    d6
216:        divu    d5,d6              * これがかなり遅くしてる。
217:        lsr.w   #1,d5
218:        scs     d3
219:        move.w  d5,d4
220:        lsr.w   #3,d5
221:        andi.w  #7,d4
222:        subq.w  #8,d4
223:        neg.w   d4
224:
225:        move.w  d4,d2              * d4 = d4*12
226:        add.w   d4,d4              *
227:        add.w   d2,d4              *
228:        add.w   d4,d4              *
229:        add.w   d4,d4              *
230:
231:        addq.w  #2,a2
232:
233:        jmp     Xline_draw(pc,d4.w)
234:                                   * glineと同じく両端から
235:                                   *        + ループ8倍展開。
236: Xline_draw:
237:        move.w  d0,(a1)+
238:        move.w  d0,-(a2)
239:        add.w   d6,d1
240:        bcc.b   Xline_11           * 68000 programmer's hand book
241:                                   * にはBcc.w はオブジェクトが2
242:                                   * バイト、クロック数が8クロッ
243:                                   * クなんて書いてあったがこれは
244:                                   * 間違いらしい。
245:        adda.w  d7,a1
246:        suba.w  d7,a2
247:                                   * glineよりe -= 2*dx の分だけ速い(はず)。
248: Xline_11:
249:        move.w  d0,(a1)+           *     8(+1)clocks( ()内はウエイト)
250:        move.w  d0,-(a2)           *     8(+1)
251:        add.w   d6,d1              *     4
252:        bcc.b   Xline_12           *    10 (?)
253:        adda.w  d7,a1              *     8
254:        suba.w  d7,a2              *     8
255:                                   * 計 30(+2) or 46(+2) clocks / 2 dots
256: Xline_12:
257:        move.w  d0,(a1)+
258:        move.w  d0,-(a2)
259:        add.w   d6,d1
260:        bcc.b   Xline_13
261:        adda.w  d7,a1
262:        suba.w  d7,a2
263: Xline_13:
264:        move.w  d0,(a1)+
265:        move.w  d0,-(a2)
266:        add.w   d6,d1
267:        bcc.b   Xline_14
268:        adda.w  d7,a1
269:        suba.w  d7,a2
270: Xline_14:
271:        move.w  d0,(a1)+
272:        move.w  d0,-(a2)
273:        add.w   d6,d1
274:        bcc.b   Xline_15
275:        adda.w  d7,a1
276:        suba.w  d7,a2
277: Xline_15:
278:        move.w  d0,(a1)+
279:        move.w  d0,-(a2)
280:        add.w   d6,d1
281:        bcc.b   Xline_16
282:        adda.w  d7,a1
283:        suba.w  d7,a2
284: Xline_16:
285:        move.w  d0,(a1)+
286:        move.w  d0,-(a2)
287:        add.w   d6,d1
288:        bcc.b   Xline_17
289:        adda.w  d7,a1
290:        suba.w  d7,a2
291: Xline_17:
292:        move.w  d0,(a1)+
293:        move.w  d0,-(a2)
294:        add.w   d6,d1
295:        bcc.b   Xline_18
296:        adda.w  d7,a1
297:        suba.w  d7,a2
298: Xline_18:
299:        dbf     d5,Xline_draw
300: l_putch1:
301:        move.w  d0,(a1)
302:        tst.b   d3
303:        beq.b   XLINE_SKIP
304: l_putch2:
305:        move.w  d0,-(a2)
306: XLINE_SKIP:
307:        rts
308: dot2:
309:        move.w  d0,(a1)
310:        move.w  d0,(a2)
311:        rts
312: YLINE:
313:        move.w  d7,d2
314:        addq.w  #2,d2
315:        swap    d5
316:        divu    d6,d5
317:        lsr.w   #1,d6    * 実は最後のバグとはここで0になる時の
318:        beq.b   dot2     * ことを考えていなかったせいでした。
319:        scs     d3
320:        move.w  d6,d4
321:        lsr.w   #3,d6
322:        andi.w  #7,d4
323:        beq.b   Yline_mod0
324:        subq.w  #8,d4
325:        neg.w   d4
326:
327:        asl.w   #4,d4    * add.w d4,d4を4回やるより
328:                         * こっちの方が2クロック速い By shadow
329:
330:        jmp     Yline_draw(pc,d4.w)
331: Yline_mod0:
332:        subq.w  #1,d6
333: Yline_draw:
334:        move.w  d0,(a1)
335:        move.w  d0,(a2)
336:        adda.w  d7,a1
337:        suba.w  d7,a2
338:        add.w   d5,d1
339:        bcc.b   Yline_11
340:        addq.w  #2,a1              * addq.w #?,An でも32ビット
341:        subq.w  #2,a2              * 操作されます。(確認済み。)
342: Yline_11:
343:        move.w  d0,(a1)            *     8(+1)
344:        move.w  d0,(a2)            *     8(+1)
345:        adda.w  d7,a1              *     8
346:        suba.w  d7,a2              *     8
347:        add.w   d5,d1              *     4
348:        bcc.b   Yline_12           *    10
349:        addq.w  #2,a1              *     4
350:        subq.w  #2,a2              *     4
351:                                   * 計 46(+2) or 54(+2) clocks / 2dots
352: Yline_12:
353:        move.w  d0,(a1)
354:        move.w  d0,(a2)
355:        adda.w  d7,a1
356:        suba.w  d7,a2
357:        add.w   d5,d1
358:        bcc.b   Yline_13
359:        addq.w  #2,a1
360:        subq.w  #2,a2
361: Yline_13:
362:        move.w  d0,(a1)
363:        move.w  d0,(a2)
364:        adda.w  d7,a1
365:        suba.w  d7,a2
366:        add.w   d5,d1
367:        bcc.b   Yline_14
368:        addq.w  #2,a1
369:        subq.w  #2,a2
370: Yline_14:
371:        move.w  d0,(a1)
372:        move.w  d0,(a2)
373:        adda.w  d7,a1
374:        suba.w  d7,a2
375:        add.w   d5,d1
376:        bcc.b   Yline_15
377:        addq.w  #2,a1
378:        subq.w  #2,a2
379: Yline_15:
380:        move.w  d0,(a1)
381:        move.w  d0,(a2)
382:        adda.w  d7,a1
383:        suba.w  d7,a2
384:        add.w   d5,d1
385:        bcc.b   Yline_16
386:        addq.w  #2,a1
387:        subq.w  #2,a2
388: Yline_16:
389:        move.w  d0,(a1)
390:        move.w  d0,(a2)
391:        adda.w  d7,a1
392:        suba.w  d7,a2
393:        add.w   d5,d1
394:        bcc.b   Yline_17
395:        addq.w  #2,a1
396:        subq.w  #2,a2
397: Yline_17:
398:        move.w  d0,(a1)
```

▶ 8月号85ページの女の子は紀子さんに似ていると思う。　佐藤　正直(18)神奈川県

```
399:            move.w  d0,(a2)
400:            add.w   d5,d1
401:            bcc.b   Yline_18
402:            adda.w  d2,a1
403:            suba.w  d2,a2
404:            dbf     d6,Yline_draw
405: l_putch3:
406:            move.w  d0,(a1)
407:            tst.b   d3
408:            beq.b   YLINE_SKIP1
409: l_putch4:
410:            move.w  d0,(a2)
411: YLINE_SKIP1:
412:            rts
413: Yline_18:
414:            adda.w  d7,a1
415:            suba.w  d7,a2
416:            dbf     d6,Yline_draw
417: l_putch5:
418:            move.w  d0,(a1)
419:            tst.b   d3
420:            beq.b   YLINE_SKIP2
421: l_putch6:
422:            move.w  d0,(a2)
423: YLINE_SKIP2:
424:            rts
425:
426: LINE45:
427:            move.w  d5,d4
428:            lsr.w   #4,d5
429:            andi.w  #15,d4
430:            subi.w  #16,d4
431:            neg.w   d4
432:
433:            add.w   d4,d4               * d4 = d4*4
434:            add.w   d4,d4               *
435:
436:            jmp     line45_draw(pc,d4.w)
437: line45_draw:
438:            move.w  d0,(a1)+            * 単なる16倍展開
439:            adda.w  d7,a1
440:            move.w  d0,(a1)+
441:            adda.w  d7,a1
442:            move.w  d0,(a1)+
443:            adda.w  d7,a1
444:            move.w  d0,(a1)+
445:            adda.w  d7,a1
446:            move.w  d0,(a1)+
447:            adda.w  d7,a1
448:            move.w  d0,(a1)+
449:            adda.w  d7,a1
450:            move.w  d0,(a1)+
451:            adda.w  d7,a1
452:            move.w  d0,(a1)+
453:            adda.w  d7,a1
454:            move.w  d0,(a1)+
455:            adda.w  d7,a1
456:            move.w  d0,(a1)+
457:            adda.w  d7,a1
458:            move.w  d0,(a1)+
459:            adda.w  d7,a1
460:            move.w  d0,(a1)+
461:            adda.w  d7,a1
462:            move.w  d0,(a1)+
463:            adda.w  d7,a1
464:            move.w  d0,(a1)+
465:            adda.w  d7,a1
466:            move.w  d0,(a1)+
467:            adda.w  d7,a1
468:            move.w  d0,(a1)+
469:            adda.w  d7,a1
470:
471:            dbf     d5,line45_draw
472: l_putch7:
473:            move.w  d0,(a1)
474:            rts
475: call_raster:
476:            exg     d2,d3
477:            jmp     raster(pc)
478:
479:            .data
480:
481: x1:
482:            dc.w    0
483: y1:
484:            dc.w    0
485: x2:
486:            dc.w    0
487: y2:
488:            dc.w    0
489: x3:
490:            dc.w    0
491: y3:
492:            dc.w    0
493:
494:            .end
495:
```

リスト3

```
  1: *
  2: * spline.s        version 1.06
  3: *
  4:            .include        magic.mac
  5:
  6:            .xdef   spline
  7:
  8:            .xref   x1
  9:            .xref   y1
 10:            .xref   x2
 11:            .xref   y2
 12:            .xref   x3
 13:            .xref   y3
 14:
 15:            .text
 16:            .even
 17:
 18: spline:
 19:            movem.w (a0)+,d1-d6
 20:            movem.w d1-d6,x1
 21:
 22:            add.w   d1,d5
 23:            asr.w   #1,d5
 24:            add.w   d3,d3
 25:            sub.w   d5,d3
 26:            move.w  d3,x2               * x2=x2*2-(x1+x3)/2
 27:            sub.w   d1,d3
 28:            swap    d3
 29:            clr.w   d3
 30:            asr.l   #3,d3   * d3.w = xa
 31:
 32:            add.w   d2,d6
 33:            asr.w   #1,d6
 34:            add.w   d4,d4
 35:            sub.w   d6,d4
 36:            move.w  d4,y2               * y2=y2*2-(y1+y3)/2
 37:            sub.w   d2,d4
 38:            swap    d4
 39:            clr.w   d4
 40:            asr.l   #3,d4   * d4.w = ya
 41:
 42:            clr.w   xa1+2
 43:            move.w  x1(pc),xa1
 44:            clr.w   ya1+2
 45:            move.w  y1(pc),ya1
 46:
 47:            move.w  x2(pc),d2
 48:            add.w   d2,d2
 49:            move.w  x1(pc),d1
 50:            sub.w   d2,d1
 51:            add.w   x3(pc),d1           * x1-x2*2+x3
 52:            move.w  d1,d2
 53:            ext.l   d2
 54:            asl.l   #8,d2
 55:            move.l  d2,a2   * move.l d2,xb
 56:            add.w   d1,d1
 57:            ext.l   d1
 58:            asl.l   #8,d1
 59:            move.l  d1,d5   * move.l d1,xc
 60:
 61:            move.w  y2(pc),d2
 62:            add.w   d2,d2
 63:            move.w  y1(pc),d1
 64:            sub.w   d2,d1
 65:            add.w   y3(pc),d1
 66:            move.w  d1,d2
 67:            ext.l   d2
 68:            asl.l   #8,d2
 69:            move.l  d2,a3   * move.l d2,yb
 70:            add.w   d1,d1
 71:            ext.l   d1
 72:            asl.l   #8,d1
 73:            move.l  d1,d6   * move.l d1,yc
 74:
 75:            move.l  a0,-(sp)
 76:            clr.l   a0
 77:            clr.l   a4
 78:            clr.l   a5
 79:            clr.l   a6
 80:            move.l  xa1(pc),d1
 81:            move.l  ya1(pc),d2
 82:
 83:            move.w  #$11-1,d7
 84:            lea.l   work(pc),a1
 85: spline2:
 86:            move.l  d1,d0
 87:            add.l   a4,d0
 88:            add.l   a6,d0
 89:            swap    d0
 90:            move.w  d0,(a1)+
 91:            move.l  d2,d0
 92:            add.l   a5,d0
 93:            add.l   a0,d0
 94:            swap    d0
 95:            move.w  d0,(a1)+
 96:
 97:            add.l   d3,a4               * xa2=xa2+xa
 98:            add.l   d4,a5               * ya2=ya2+ya
 99:            move.l  a2,d0
100:            add.l   d0,a6               * xa3=xa3+xb
101:            add.l   d5,a2               * xb=xb+xc
102:            move.l  a3,d0
103:            add.l   d0,a0               * ya3=ya3+yb
104:            add.l   d6,a3               * yb=yb+yc
105:
106:            dbf     d7,spline2
107:
108:            lea.l   spline_data(pc),a0
109:            MAGIC   __LINE
110:            move.l  (sp)+,a0
111:            rts
112:
113: spline_data:
114:            dc.w    17
115: work:
116:            ds.b    68
117:
118:            .data
119: xa1:
120:            ds.b    3
121:            ds.b    1
122: ya1:
123:            ds.b    3
124:            ds.b    1
125:
126:            .end
127:
```

リスト4

```
  1: *
  2: * box.s
  3: *
  4:            .xdef   box
  5:            .xdef   b_gbase
  6:            .xdef   b_sft_time
  7:            .xdef   b_pixel
  8:
  9:            .xref   raster_main
 10:            .xref   color
 11:            .xref   minx
 12:            .xref   maxx
 13:            .xref   miny
 14:            .xref   maxy
 15:
 16:            .text
 17:            .even
 18:
 19: box:
 20:            movem.w (a0)+,d1-d4
 21: cliping:
 22:            cmp.w   d1,d3
 23:            bge     clip0
 24:            exg.l   d1,d3
 25: clip0:                             * y1≧y2にする
 26:            cmp.w   d2,d4
 27:            bge     b_pixel
 28:            exg.l   d2,d4
 29: b_pixel:
 30:            move.w  #1024,d6
```

▶ 8月号80ページの福原徹氏が描いた“Miss GEOS”を見て絶句しました。どうしてあのように描けるのでしょうか。リアルすぎてポスターにしたいほどです（特に髪の毛1本1本やGパンの質感など）。氏は女性の体を知りつくしている？　　林田　和也(19)千葉県

```
 31:         move.w  d6,d5
 32:                         * クリッピング
 33:         cmp.w   minx(pc),d3
 34:         blt     clip_end
 35:         cmp.w   minx(pc),d1
 36:         bge     clip2
 37:         move.w  minx(pc),d1
 38:         clr.w   d5      * offset 0
 39: clip2:
 40:         cmp.w   maxx(pc),d1
 41:         bgt     clip_end
 42:         cmp.w   maxx(pc),d3
 43:         ble     clip3
 44:         move.w  maxx(pc),d3
 45:         clr.w   d6      * offset 0
 46: clip3:
 47:         cmp.w   miny(pc),d4
 48:         blt     clip_end
 49:         cmp.w   miny(pc),d2
 50:         bge     clip4
 51:         move.w  miny(pc),d2
 52: clip4:
 53:         cmp.w   maxy(pc),d2
 54:         bgt     clip_end
 55:         cmp.w   maxy(pc),d4
 56:         ble     clip_end
 57:         move.w  maxy(pc),d4
 58: clip_end:
 59:         move.w  d1,a5   * ワークにコピー
 60:         move.w  d3,a6   * ワークにコピー
 61:         move.w  d4,d7
 62:         sub.w   d2,d7   * ライン数－1
 63:         sub.w   d1,d3
 64:         exg     d1,d3
 65:         addq.w  #1,d1   * 1ラインのドット数
 66: b_gbase:
 67:         lea.l   $c00002,a1      * gram top address+2
 68:
 69: * プリデクリメントを使うから ＋2 してある
 70:
 71: b_sft_time:
 72:         moveq.l #10,d0  * シフトカウント
 73:         asl.l   d0,d2   * d0を1024(2048)倍する
 74:         asl.l   d0,d4
 75:         movea.l a1,a3
 76:         adda.l  d4,a3
 77:         adda.w  a6,a3
 78:         adda.w  a6,a3   * (x2+1,y2)
 79:         adda.l  d2,a1
 80:         movea.l a1,a2
 81:         adda.w  a5,a2
 82:         adda.w  a5,a2   * (x1+1,y1)
 83:         adda.w  a6,a1
 84:         adda.w  a6,a1   * (x2+1,y1)
 85:         move.l  a1,-(sp)
 86:         move.l  a2,-(sp)
 87:         move.l  d1,-(sp)
 88:         jsr     raster_main(pc)
 89:         move.l  (sp)+,d1
 90:         movea.l a3,a1
 91:         jsr     raster_main(pc)
 92:         move.l  (sp)+,a2
 93:         move.l  (sp)+,a1
 94:         subq.w  #2,a1
 95:         subq.w  #2,a2
 96:         move.w  color(pc),d0
 97:
 98:         move.w  d7,d4
 99:         asr.w   #3,d7
100:         andi.w  #7,d4
101:         subq.w  #8,d4
102:         neg.w   d4
103:         asl.w   #3,d4
104:         jmp     b_yloop(pc,d4.w)
105: b_yloop:
106:         move.w  d0,(a1)
107:         move.w  d0,(a2)
108:         adda.w  d6,a1
109:         adda.w  d5,a2
110:         move.w  d0,(a1)
111:         move.w  d0,(a2)
112:         adda.w  d6,a1
113:         adda.w  d5,a2
114:         move.w  d0,(a1)
115:         move.w  d0,(a2)
116:         adda.w  d6,a1
117:         adda.w  d5,a2
118:         move.w  d0,(a1)
119:         move.w  d0,(a2)
120:         adda.w  d6,a1
121:         adda.w  d5,a2
122:         move.w  d0,(a1)
123:         move.w  d0,(a2)
124:         adda.w  d6,a1
125:         adda.w  d5,a2
126:         move.w  d0,(a1)
127:         move.w  d0,(a2)
128:         adda.w  d6,a1
129:         adda.w  d5,a2
130:         move.w  d0,(a1)
131:         move.w  d0,(a2)
132:         adda.w  d6,a1
133:         adda.w  d5,a2
134:         move.w  d0,(a1)
135:         move.w  d0,(a2)
136:         adda.w  d6,a1
137:         adda.w  d5,a2
138:         dbf     d7,b_yloop
139:         rts
140:
141:         .end
```

リスト5

```
  1: *
  2: * triangle.s
  3: *
  4:         .xdef   triangle
  5:         .xref   raster_pset
  6:
  7:         .xref   x1
  8:         .xref   y1
  9:         .xref   x2
 10:         .xref   y2
 11:         .xref   x3
 12:         .xref   y3
 13:
 14:         .text
 15:         .even
 16:
 17: triangle:
 18:         movem.w (a0)+,d1-d6
 19:
 20:         move.l  a0,-(sp)
 21: **************************************
 22: * 頂点の座標(x1,y1)(x2,y2)(x3,y3)を
 23: *      y1≦y2≦y3
 24: * の関係を満たすようにする
 25: **************************************
 26:         cmp.w   d4,d6
 27:         bge     triangle2       * y3≧y2  --変更--
 28:         exg     d3,d5
 29:         exg     d4,d6
 30: triangle2:
 31:         cmp.w   d2,d6
 32:         bge     triangle3       * y3≧y1  --変更--
 33:         exg     d2,d6
 34:         exg     d2,d4
 35:         exg     d1,d5
 36:         exg     d1,d3
 37: triangle3:
 38:         cmp.w   d2,d4
 39:         bge     triangle4       * y2≧y1  --変更--
 40:         exg     d1,d3
 41:         exg     d2,d4
 42:
 43: ********************************************
 44: * a2.w=sx , a3.w=ex , a4.w=r_1 , a5.w=r_2 *
 45: * a6.w=py
 46: ********************************************
 47: triangle4:
 48:         movem.w d1-d6,x1
 49:         move.w  d1,a2           * move.w d1,sx
 50:         move.w  d1,a3           * move.w d1,ex
 51:         clr.w   r_1             * clr.w r_1
 52:         clr.w   r_2             * clr.w r_2
 53:         move.w  #$5240,triangle14       * addq.w #1,d0
 54:         sub.w   d1,d3
 55:         beq     triangle4_1     * x2-x1=0
 56:         bpl     triangle5       * x2-x1>0
 57:         neg.w   d3
 58:         move.w  #$5340,triangle14       * subq.w #1,d0
 59:         bra     triangle5
 60: triangle4_1:
 61:         move.w  #$4e71,triangle14       * nop
 62: triangle5:
 63:         move.w  d3,a4
 64:
 65:         move.w  #$5240,triangle18       * addq.w #1,d0
 66:         sub.w   d1,d5
 67:         beq     triangle5_1     * x3-x1=0
 68:         bpl     triangle6       * x3-x1>0
 69:         neg.w   d5
 70:         move.w  #$5340,triangle18       * subq.w #1,d0
 71:         bra     triangle6
 72: triangle5_1:
 73:         move.w  #$4e71,triangle18       * move.w #$4e71,d0  $4e71 = nop
 74: triangle6:
 75:         move.w  d5,a5
 76:
 77:         move.w  y1(pc),py
 78:         sub.w   d2,d6           * y3-y1
 79:         move.w  d6,a6
 80:
 81:         move.w  y3(pc),d6
 82:         sub.w   d4,d6           * d6.w = y3-y2
 83:         sub.w   d2,d4           * d4.w = y2-y1
 84:         move.w  d4,d5
 85:         move.w  d5,a0
 86:         tst.w   d5
 87:         bne     triangle7
 88:         move.w  x2(pc),a2
 89:         tst.w   d6
 90:         bne     triangle8
 91:                                 * y1=y2=y3
 92:         move.w  x1(pc),d1
 93:         move.w  x2(pc),d2
 94:         move.w  y1(pc),d3
 95:         jsr     raster_pset(pc)
 96:         move.w  x2(pc),d1
 97:         move.w  x3(pc),d2
 98:         move.w  y1(pc),d3
 99:         jsr     raster_pset(pc)
100:         bra     triangle_end
101: triangle7:
102:         move.w  d5,d1
103:         asr.w   #1,d1
104:         move.w  d1,r_1          * move.w d1,r_1
105:         move.w  a6,d1
106:         asr.w   #1,d1
107:         move.w  d1,r_2          * move.w d1,r_2
108:         subq.w  #1,d5
109:         jsr     triangle10(pc)
110:         tst.w   d6
111:         beq     triangle_end
112: triangle8:
113:         move.w  x2(pc),a2
114:         move.w  d6,a0
115:         move.w  d6,d5
116:         asr.w   #1,d6
117:         move.w  d6,r_1          * move.w d6,r_1
118:         move.w  x3(pc),d1
119:         move.w  #$5240,triangle14       * addq.w #1,d0
120:         sub.w   x2(pc),d1
121:         beq     triangle8_1
122:         bpl     triangle9
123:         neg.w   d1
124:         move.w  #$5340,triangle14       * subq.w #1,d0
125:         bra     triangle9
126: triangle8_1:
127:         move.w  #$4e71,triangle14       * nop
128: triangle9:
129:         move.w  d1,a4
130:
131:         jsr     triangle10(pc)
132: triangle_end:
133:         move.l  (sp)+,a0
134:         rts
135:
136: triangle10:
137:         move.w  py(pc),d3
138:         move.w  r_1(pc),d4
139:         move.w  r_2(pc),d7
140: triangle11:
141:         move.w  a2,d1
142:         move.w  a3,d2
143:         jsr     raster_pset(pc)
144:         addq.w  #1,d3
145:         sub.w   a4,d4
146:         bpl     triangle15
147:         move.w  a2,d0   * move.w sx,d0
148: triangle14:
149:         nop                     * x2>x1なら addq.w #1,d0
150:                                 * x2=x1なら nop
151:                                 * x2<x1なら subq.w #1,d0
152:         add.w   a0,d4
153:         bcc     triangle14
154:         move.w  d0,a2   * move.w d0,sx
```

▶ 8月号80,81ページのなんて贅沢なこと。いつもなら大量のデータを載っけているのに。
主藤　二裕(23)福岡県

```
155: triangle15:
156:         sub.w   a5,d7
157:         bpl     triangle20
158:         move.w  a3,d0   * move.w ex,d0
159: triangle18:
160:         nop                             * x2>x1なら addq.w #1,d0
161:                                         * x2=x1なら nop
162:                                         * x2<x1なら subq.w #1,d0
163: triangle19:
164:         add.w   a6,d7
165:         bcc     triangle18
166:         move.w  d0,a3   * move.w d0,ex
167: triangle20:
168:         dbf     d5,triangle11
169:         move.w  d3,py
170:         rts
171:
172:
173:         .data
174:
175: r_1:
176:         ds.w    1
177: r_2:
178:         ds.w    1
179: py:
180:         ds.w    1
181:
         .end
```

リスト6

```
  1: *
  2: * boxfull.s
  3: *
  4:         .xdef   boxfull
  5:         .xdef   f_gbase
  6:         .xdef   f_pixel
  7:         .xdef   f_sft_time
  8:
  9:         .xref   raster_main
 10:         .xref   minx
 11:         .xref   maxx
 12:
 13:         .text
 14:         .even
 15:
 16: boxfull:
 17:         movem.w (a0)+,d1-d4     * x0,y0
 18:                                 * x1,y1
 19:         cmp.w   d1,d3
 20:         bpl     skip1
 21:         exg     d1,d3
 22: skip1:
 23:         cmp.w   d2,d4
 24:         bpl     clip_minx
 25:         exg     d2,d4
 26: clip_minx:                      * クリッピング処理を追加
 27:         cmp.w   minx(pc),d1     *                 by shadow
 28:         bge     clip_maxx       *
 29:         cmp.w   minx(pc),d3     *
 30:         blt     boxfull_end     *
 31:         move.w  minx(pc),d1     *
 32: clip_maxx:                      *
 33:         cmp.w   maxx(pc),d3     *
 34:         ble     clip_end        *
 35:         cmp.w   maxx(pc),d1     *
 36:         bgt     boxfull_end     *
 37:         move.w  maxx(pc),d3     *
 38: clip_end:
 39:         move.w  d3,d5
 40:         sub.w   d1,d5   * d5.w = x2-x1
 41:         addq.w  #1,d5   * bug fix 0でも1ドットある
 42:         sub.w   d2,d4   * d4.w = y2-y1
 43:         addq.w  #1,d4   * bug fix 0でも1ラインある
 44:
 45: f_gbase:
 46:         move.l  #$c00000+2,d6
 47: f_pixel:
 48:         move.w  #1024,d7
 49:         ext.l   d7
 50: f_sft_time:
 51:         moveq.l #10,d0
 52:         lsl.l   d0,d2
 53:         add.w   d3,d3
 54:         add.w   d3,d2
 55:         add.l   d2,d6   * d6.l = vram address
 56:
 57:         move.w  d4,d3
 58:         asr.w   #3,d4
 59:         andi.w  #7,d3
 60:         neg.w   d3
 61:         addq.w  #8,d3
 62:         move.w  d3,d2
 63:         add.w   d3,d3
 64:         add.w   d3,d3
 65:         add.w   d2,d3
 66:         add.w   d3,d3
 67:         jmp     loop(pc,d3.w)
 68: loop:
 69:         move.w  d5,d1
 70:         movea.l d6,a1
 71:         jsr     raster_main(pc)
 72:         add.l   d7,d6
 73:         move.w  d5,d1
 74:         movea.l d6,a1
 75:         jsr     raster_main(pc)
 76:         add.l   d7,d6
 77:         move.w  d5,d1
 78:         movea.l d6,a1
 79:         jsr     raster_main(pc)
 80:         add.l   d7,d6
 81:         move.w  d5,d1
 82:         movea.l d6,a1
 83:         jsr     raster_main(pc)
 84:         add.l   d7,d6
 85:         move.w  d5,d1
 86:         movea.l d6,a1
 87:         jsr     raster_main(pc)
 88:         add.l   d7,d6
 89:         move.w  d5,d1
 90:         movea.l d6,a1
 91:         jsr     raster_main(pc)
 92:         add.l   d7,d6
 93:         move.w  d5,d1
 94:         movea.l d6,a1
 95:         jsr     raster_main(pc)
 96:         add.l   d7,d6
 97:         move.w  d5,d1
 98:         movea.l d6,a1
 99:         jsr     raster_main(pc)
100:         add.l   d7,d6
101:
102:         dbf     d4,loop
103: boxfull_end:
104:         rts
105:
106:         .end
107:
```

リスト7

```
  1: *
  2: * circle.s
  3: *
  4:         .xdef   circle
  5:         .xdef   c_gbase
  6:         .xdef   c_pixel
  7:         .xdef   c_sft_time
  8:
  9:         .xref   raster_pset
 10:         .xref   raster_main
 11:         .xref   color
 12:         .xref   minx
 13:         .xref   miny
 14:         .xref   maxx
 15:         .xref   maxy
 16:
 17:         .text
 18:         .even
 19:
 20: circle:
 21:         movem.w (a0)+,d1-d3
 22:
 23:         move.w  d1,d4
 24:         move.w  d2,d5
 25:
 26:         move.w  d1,d6
 27:         move.w  d2,d7
 28:         sub.w   d3,d1
 29:         sub.w   d3,d2
 30:         add.w   d3,d6
 31:         add.w   d3,d7
 32:
 33:         cmp.w   minx(pc),d1
 34:         bge     sk1
 35:         cmp.w   minx(pc),d6
 36:         blt     cir_end
 37:         bra     clip_circle
 38: sk1:
 39:         cmp.w   miny(pc),d2
 40:         bge     sk2
 41:         cmp.w   miny(pc),d7
 42:         blt     cir_end
 43:         bra     clip_circle
 44: sk2:
 45:         cmp.w   maxx(pc),d6
 46:         ble     sk3
 47:         cmp.w   maxx(pc),d1
 48:         bgt     cir_end
 49:         bra     clip_circle
 50: sk3:
 51:         cmp.w   maxy(pc),d7
 52:         ble     sk4
 53:         cmp.w   maxy(pc),d2
 54:         bgt     cir_end
 55:         bra     clip_circle
 56: sk4:
 57:         move.w  d4,d1
 58:         move.w  d5,d2
 59:
 60: c_gbase:
 61:         lea.l   $c00000,a3      * gram base address
 62:
 63:         move.w  d3,d4
 64: c_pixel:
 65:         move.w  #1024,d7
 66: c_sft_time:
 67:         moveq.l #10,d5
 68:         asl.l   d5,d2
 69:         asl.l   d5,d4
 70:         add.w   d1,d1
 71:         add.w   d1,d2
 72:         adda.l  d2,a3
 73:         tst.w   d3
 74:         beq     dot
 75:         movea.l a3,a4
 76:         movea.l a3,a6
 77:         add.w   d3,d3
 78:         suba.l  d4,a3
 79:         adda.l  d4,a6
 80:         adda.l  d3,a4
 81:         movea.l a4,a5
 82: non_clip_circle:
 83:         clr.w   d5                      * 2x
 84:         move.w  d3,d6                   * 2y
 85:         clr.w   d4                      * e
 86:
 87:         move.w  d3,d1
 88:         movea.l a4,a1
 89:         jsr     raster_main(pc)
 90: ncc_loop:
 91:         move.w  d4,d0
 92:         add.w   d5,d0   *       e1
 93:         addq.w  #1,d1
 94:         move.w  d0,d2
 95:         bpl     ncc1
 96:         neg.w   d2
 97: ncc1:
 98:         move.w  d0,d1
 99:         sub.w   d6,d1   *       e2
100:         addq.w  #1,d1
101:         move.w  d1,d3
102:         bpl     ncc2
103:         neg.w   d3
104: ncc2:
105:         cmp.w   d2,d3
106:         bge     ncc4    * if |e1|<|e2| then ncc4
107: ncc3:
108:         move.w  d1,d4
109:         move.w  d5,d1
110:         movea.l a3,a1
111:         jsr     raster_main(pc)
112:         move.w  d5,d1
113:         movea.l a6,a1
114:         jsr     raster_main(pc)
115:         subq.w  #2,d6
116:         subq.w  #2,a4
117:         subq.w  #2,a5
118:         adda.w  d7,a3
119:         suba.w  d7,a6
120:         cmp.w   d5,d6
121:         bgt     ncc5
122:         rts
```

▶今月の特集はなかなか面白かった。いままでプリンタは買わないつもりだったんだけど
すぐにでもほしくなってしまった。やはり印刷はいいなあ。　原田　通男(17)埼玉県

```
123: ncc4:
124:         move.w  d0,d4
125: ncc5:
126:         addq.w  #2,d5
127:         addq.w  #2,a3
128:         addq.w  #2,a6
129:         suba.w  d7,a4
130:         adda.w  d7,a5
131:         move.w  d6,d1
132:         movea.l a4,a1
133:         jsr     raster_main(pc)
134:         move.w  d6,d1
135:         movea.l a5,a1
136:         jsr     raster_main(pc)
137:
138:         cmp.w   d5,d6
139:         bgt     ncc_loop
140: cir_end:
141:         rts
142:
143: clip_circle:
144:         move.w  d3,d6           * y
145:         move.w  d4,d2           * Ox
146:         move.w  d5,d3           * Oy
147:         clr.w   d5              * x
148:         clr.w   d7              * e
149:
150:         swap    d5
151:         swap    d6
152:         move.w  d2,d1
153:         move.w  d2,d5
154:         move.w  d3,d6
155:         swap    d5
156:         swap    d6
157:
158:         add.w   d6,d1
159:         sub.w   d6,d2
160:         jsr     raster_pset(pc)
161: cc_loop:
162:         move.w  d5,d0
163:         add.w   d0,d0
164:         addq.w  #1,d0
165:         add.w   d7,d0   *       e1
166:         move.w  d0,d2
167:         bpl     cc1
168:         neg.w   d2
169: cc1:
170:         move.w  d6,d1
171:         add.w   d1,d1
172:         neg.w   d1
173:         addq.w  #1,d1
174:         add.w   d0,d1   *       e2
175:         move.w  d1,d3
176:         bpl     cc2
177:         neg.w   d3
178: cc2:
179:         cmp.w   d2,d3
180:         bge     cc4     * if |e1|<|e2| then cc4
181: cc3:
182:         move.w  d1,d7
183:
184:         swap    d5
185:         swap    d6
186:         move.w  d5,d1
187:         move.w  d5,d2
188:         move.w  d6,d3
189:         move.w  d6,d4
190:         swap    d5
191:         swap    d6
192:         add.w   d5,d1
193:         sub.w   d5,d2
194:         add.w   d6,d3
195:         sub.w   d6,d4
196:         jsr     raster_pset(pc)
197:         move.w  d4,d3
198:         jsr     raster_pset(pc)
199:         subq.w  #1,d6
200:         cmp.w   d5,d6
201:         bgt     cc5
202:         rts
203: cc4:
204:         move.w  d0,d7
205: cc5:
206:         addq.w  #1,d5
207:         swap    d5
208:         swap    d6
209:         move.w  d5,d1
210:         move.w  d5,d2
211:         move.w  d6,d3
212:         move.w  d6,d4
213:         swap    d5
214:         swap    d6
215:         add.w   d6,d1
216:         sub.w   d6,d2
217:         add.w   d5,d3
218:         sub.w   d5,d4
219:
220:         jsr     raster_pset(pc)
221:         move.w  d4,d3
222:         jsr     raster_pset(pc)
223:
224:         cmp.w   d5,d6
225:         bgt     cc_loop
226:         rts
227: dot:
228:         move.w  color(pc),(a3)
229:         rts
230:
231:         .end
```

リスト8

```
 1: *
 2: * window.s
 3: *
 4:         .xdef   window
 5:         .xdef   minx
 6:         .xdef   miny
 7:         .xdef   maxx
 8:         .xdef   maxy
 9:
10:         .xref   point_ox1
11:         .xref   point_oy1
12:         .xref   point_ox2
13:         .xref   point_oy2
14:
15:         .text
16:         .even
17:
18: window:
19:         movem.w (a0)+,d1-d4
20:         movem.w d1-d4,minx
21:
22:         move.w  minx(pc),d1
23:         add.w   maxx(pc),d1
24:         asr.w   #1,d1
25:         move.w  d1,point_ox1+2
26:         move.w  d1,point_ox2+2
27:
28:         move.w  miny(pc),d1
29:         add.w   maxy(pc),d1
30:         asr.w   #1,d1
31:         move.w  d1,point_oy1+2
32:         move.w  d1,point_oy2+2
33:         rts
34:
35:         .data
36:
37: minx:
38:         ds.w    1
39: miny:
40:         ds.w    1
41: maxx:
42:         ds.w    1
43: maxy:
44:         ds.w    1
45:
46:         .end
47:
```

リスト9

```
  1: *
  2: * mode.s
  3: *
  4:         .xdef   mode
  5:         .xdef   disp_mode
  6:
  7:         .xref   Xline_draw
  8:         .xref   Yline_draw
  9:         .xref   line45_draw
 10:         .xref   dot2
 11:         .xref   l_putch1
 12:         .xref   l_putch2
 13:         .xref   l_putch3
 14:         .xref   l_putch4
 15:         .xref   l_putch5
 16:         .xref   l_putch6
 17:         .xref   l_putch7
 18:         .xref   ras_mode1
 19:         .xref   ras_mode2
 20:         .xref   r_dot
 21:
 22:         .text
 23:         .even
 24:
 25: mode:
 26:         move.w  (a0)+,d1
 27:         move.w  d1,disp_mode    * ラインモード
 28:         lea.l   mode_tbl(pc),a1
 29:         add.w   d1,d1
 30:         move.w  d1,d2
 31:         add.w   d1,d1
 32:         add.w   d1,d1
 33:         add.w   d2,d1
 34:         movem.w (a1,d1.w),d2-d6
 35:
 36:         lea.l   Xline_draw(pc),a1
 37:         moveq.l #12,d1
 38:         move.w  #8-1,d0
 39: mode_loop1:
 40:         move.w  d4,0(a1)
 41:         move.w  d5,2(a1)
 42:         adda.l  d1,a1
 43:         dbf     d0,mode_loop1
 44:         move.w  d2,l_putch1
 45:         move.w  d5,l_putch2
 46:         move.w  d2,dot2
 47:         move.w  d3,dot2+2
 48:
 49:         lea.l   Yline_draw(pc),a1
 50:         moveq.l #16,d1
 51:         move.w  #8-1,d0
 52: mode_loop2:
 53:         move.w  d2,0(a1)
 54:         move.w  d3,2(a1)
 55:         adda.l  d1,a1
 56:         dbf     d0,mode_loop2
 57:         move.w  d2,l_putch3
 58:         move.w  d3,l_putch4
 59:         move.w  d2,l_putch5
 60:         move.w  d3,l_putch6
 61:
 62:         lea.l   line45_draw(pc),a1
 63:         move.w  #16-1,d0
 64: mode_loop3:
 65:         move.w  d4,(a1)+
 66:         addq.w  #2,a1
 67:         dbf     d0,mode_loop3
 68:         move.w  d2,l_putch7
 69:
 70:         move.w  d5,ras_mode1
 71:         move.w  d6,ras_mode2
 72:         move.w  d5,r_dot
 73:         rts
 74:
 75:         .data
 76:
 77: mode_tbl:
 78:                         *preset_data
 79:         dc.w    $4251   * clr.w (a1)
 80:         dc.w    $4252   * clr.w (a2)
 81:         dc.w    $4259   * clr.w (a1)+
 82:         dc.w    $4262   * clr.w -(a2)
 83:         dc.w    $42a2   * clr.l -(a2)
 84:                         *xor_data
 85:         dc.w    $b151   * eor.w d0,(a1)
 86:         dc.w    $b152   * eor.w d0,(a2)
 87:         dc.w    $b159   * eor.w d0,(a1)+
 88:         dc.w    $b162   * eor.w d0,-(a2)
 89:         dc.w    $b1a2   * eor.l d0,-(a2)
 90:                         *pset_data
 91:         dc.w    $3280   * move.w d0,(a1)
 92:         dc.w    $3480   * move.w d0,(a2)
 93:         dc.w    $32c0   * move.w d0,(a1)+
 94;         dc.w    $3500   * move.w d0,-(a2)
 95:         dc.w    $2500   * move.l d0,-(a2)
 96:                         *nop_data
 97:         dc.w    $4e71   * nop
 98:         dc.w    $4e71   * nop
 99:         dc.w    $4e71   * nop
100:         dc.w    $4e71   * nop
101:         dc.w    $4e71   * nop
102:
103: disp_mode:
104:         ds.w    1
105:
106:         .end
```


▶8月号167ページの田中さん,もっと種類があります。レトルトパックの「シーフードめし」ってのもあるし,箱パックになっているカレーもあるんだぞ。ちなみに「つけもの」などのおかずも缶詰です。
棟方　正治(23)青森県

リスト 10

```
 1: *
 2: * wipe.s
 3: *
 4:         .xdef   cls
 5:         .xdef   wipe_putch
 6:
 7:         .xref   boxfull
 8:         .xref   f_gbase
 9:         .xref   minx
10:         .xref   color
11:         .xref   page
12:
13:         .text
14:         .even
15:
16: cls:
17:         move.l  a0,-(sp)
18:
19:         tst.b   page
20:         bne     wipe_putch
21:         bclr.b  #3,f_gbase+3    * vram top address $c00000
22:         bra     cls2
23: wipe_putch:
24:         bset.b  #3,f_gbase+3    * vram top address $c80000
25: cls2:
26:         move.w  color(pc),-(sp)
27:         clr.w   color
28:         lea.l   minx(pc),a0
29:         jsr     boxfull(pc)
30:         move.w  (sp)+,color
31:
32:         move.l  (sp)+,a0
33:         rts
34:
35:         .end
```

リスト 11

```
 1: *
 2: * parameter.s   version 1.30
 3: *
 4:         .xdef   para
 5:         .xref   cx
 6:
 7:         .text
 8:         .even
 9:
10: para:
11:         move.w  (a0)+,d0
12:         add.w   d0,d0
13:         lea.l   cx(pc),a1
14:         move.w  (a0)+,(a1,d0.w)
15:         rts
16:
17:         .end
18:
```

リスト 12

```
 1: *
 2: * data.s        version 1.31
 3: *
 4:         .xdef   data
 5:         .xdef   data_mode
 6:         .xdef   obj_no
 7:         .xdef   obj_no2
 8:
 9:         .xref   pct_work
10:         .xref   obj_adr
11:         .xref   line_adr
12:         .xref   point_adr
13:         .xref   data_adr
14:         .xref   pct
15:         .xref   pct2
16:         .xref   page
17:
18:         .text
19:         .even
20:
21: data:
22:         tst.b   page
23:         bne     data1
24:                                 * ページ0
25:         lea.l   pct(pc),a2
26:         lea.l   obj_no(pc),a3
27:         bra     data2
28: data1:                          * ページ1
29:         lea.l   pct2(pc),a2
30:         lea.l   obj_no2(pc),a3
31: data2:
32:         move.w  (a0)+,d0
33:         ext.l   d0
34:         move.w  d0,pct_work     * PCT
35:         add.w   d0,(a2)
36:
37:         move.l  a0,obj_adr
38:
39:         add.w   d0,d0
40:         move.w  d0,d3
41:         add.w   d0,d0
42:         add.w   d3,d0           * PCT*6
43:         adda.l  d0,a0
44:
45:         move.l  line_adr(pc),a4
46:         move.w  (a0)+,d0        * LCT
47:         move.w  d0,d3           * d0をワークにコピー
48:
49:         move.l  data_adr(pc),a2
50:         move.l  point_adr(pc),(a2)+
51:                                 * 2D変換後の座標データ
52:         move.w  d0,(a2)+        * 物体の線分数
53: data_mode:
54:         move.w  (a0)+,(a2)+     * 拡張モードでは物体のカラーコード
55:                                 * コンパチモードでは
56:                                 *  NOP が書き込まれる
57:         move.l  a2,data_adr
58:
59:         asr.w   #4,d3           * d3=d3/16 … カウンタ
60:         andi.w  #15,d0
61:         neg.w   d0
62:         add.w   #16,d0
63:         add.w   d0,d0
64:         jmp     data3(pc,d0.w)
65: data3:
66:         move.l  (a0)+,(a4)+
67:         move.l  (a0)+,(a4)+
68:         move.l  (a0)+,(a4)+
69:         move.l  (a0)+,(a4)+
70:         move.l  (a0)+,(a4)+
71:         move.l  (a0)+,(a4)+
72:         move.l  (a0)+,(a4)+
73:         move.l  (a0)+,(a4)+
74:         move.l  (a0)+,(a4)+
75:         move.l  (a0)+,(a4)+
76:         move.l  (a0)+,(a4)+
77:         move.l  (a0)+,(a4)+
78:         move.l  (a0)+,(a4)+
79:         move.l  (a0)+,(a4)+
80:         move.l  (a0)+,(a4)+
81:         move.l  (a0)+,(a4)+
82:         dbf     d3,data3
83: data_end:
84:         move.l  a4,line_adr
85:         addq.w  #1,(a3)         * 登録物体数を増やす
86:         rts
87:
88:         .data
89:
90: obj_no:
91:         ds.w    1               * ページ0に登録した物体の数
92: obj_no2:
93:         ds.w    1               * ページ1に登録した物体の数
94:
95:         .end
96:
97:
98:
```

リスト 13

```
  1: *
  2: * perspective.s           version 1.31
  3: *
  4:         .xdef   perspective
  5:         .xdef   pct
  6:         .xdef   pct2
  7:         .xdef   pct_work
  8:         .xdef   cx
  9:
 10:         .xdef   obj_adr
 11:         .xdef   line_adr
 12:         .xdef   point_adr
 13:         .xdef   data_adr
 14:
 15:         .xdef   disp_buf
 16:         .xdef   disp_buf2
 17:         .xdef   lin_buf
 18:         .xdef   lin_buf2
 19:         .xdef   data_buf
 20:         .xdef   data_buf2
 21:
 22:         .xdef   point_ox1
 23:         .xdef   point_oy1
 24:         .xdef   p_end
 25:
 26:         .xref   page
 27:         .xref   sinhl
 28:         .xref   coshl
 29:         .xref   minz
 30:
 31:         .text
 32:         .even
 33:
 34: perspective:
 35:         move.l  a0,-(sp)
 36:         tst.w   pct_work
 37:         beq     perspective_end
 38:
 39:         lea.l   cosh(pc),a1
 40:         movem.w head(pc),a2-a4
 41:         move.w  a2,d1
 42:         jsr     coshl(pc)
 43:         move.w  a2,d1
 44:         jsr     sinhl(pc)
 45:         move.w  a3,d1
 46:         jsr     coshl(pc)
 47:         move.w  a3,d1
 48:         jsr     sinhl(pc)
 49:         move.w  a4,d1
 50:         jsr     coshl(pc)
 51:         move.w  a4,d1
 52:         jsr     sinhl(pc)
 53:         movem.w cosh(pc),d0-d5
 54:         move.w  d0,a1
 55:         move.w  d1,a2
 56:         move.w  d4,a3
 57:         move.w  d5,a4
 58:
 59: * cosh sinh cosp sinp cosb sinb
 60: *   d0   d1   d2   d3   d4   d5
 61:
 62:         move.w  d3,d6
 63:         muls    d1,d6
 64:         add.l   d6,d6
 65:         add.l   d6,d6
 66:         swap    d6
 67:         muls    d6,d5           * sinb破壊
 68:         muls    d4,d0           * cosh破壊
 69:         add.l   d5,d0
 70:         add.l   d0,d0
 71:         add.l   d0,d0
 72:         swap    d0
 73:         move.w  d0,pers2+2      * pers2+2=sinh*sinp*sinb+cosh*cosb
 74:         move.w  a1,d0           * cosh復帰
 75:         move.w  a4,d5           * sinb復帰
 76:         muls    d5,d0
 77:         muls    d6,d4
 78:         sub.l   d0,d4
 79:         add.l   d4,d4
 80:         add.l   d4,d4
 81:         swap    d4
 82:         move.w  d4,pers6+2      * pers6+2=sinh*sinp*cosb-sinb*cosh
 83:         move.w  a1,d0           * cosh復帰
 84:         move.w  a3,d4           * cosb復帰
 85:
 86: * cosh sinh cosp sinp cosb sinb
 87: *   d0   d1   d2   d3   d4   d5
 88:
 89:         move.w  d3,d6
 90:         muls    d0,d6
 91:         add.l   d6,d6
 92:         add.l   d6,d6
 93:         swap    d6
 94:         muls    d6,d5           * sinb破壊
 95:         muls    d4,d1           * sinh破壊
 96:         sub.l   d1,d5
 97:         add.l   d5,d5
 98:         add.l   d5,d5
 99:         swap    d5
100:         move.w  d5,pers4+2      * pers4+2=cosh*sinp*sinb-sinh*cosb
```

▶いま,UFOキャッチャーのアンパンマンシリーズを集めることがひそかなブームだ。
稲寺　健至(20)神奈川県

```
101:            move.w  a2,d1               * sinh復帰
102:            move.w  a4,d5               * sinb復帰
103:            muls    d5,d1               * sinh破壊
104:            muls    d6,d4               * cosb破壊
105:            add.l   d1,d4
106:            add.l   d4,d4
107:            add.l   d4,d4
108:            swap    d4
109:            move.w  d4,pers8+2          * pers8+2=cosh*sinp*cosb+sinh*sinb
110:            move.w  a2,d1               * sinh復帰
111:            move.w  a3,d4               * cosb復帰
112:
113: * cosh sinh cosp sinp cosb sinb
114: *   d0   d1   d2   d3   d4   d5
115:
116:            muls    d2,d1               * sinh破壊
117:            add.l   d1,d1
118:            add.l   d1,d1
119:            swap    d1
120:            move.w  d1,pers10+2         * pers10+2=sinh*cosp
121:            muls    d2,d5               * sinb破壊
122:            add.l   d5,d5
123:            add.l   d5,d5
124:            swap    d5
125:            move.w  d5,pers3+2          * pers3+2=sinb*cosp
126:            muls    d2,d4               * cosb破壊
127:            add.l   d4,d4
128:            add.l   d4,d4
129:            swap    d4
130:            move.w  d4,pers7+2          * pers7+2=cosp*cosb
131:            neg.w   d3                  * sinp破壊
132:            move.w  d3,pers11+2         * pers11+2=neg(sinp)
133:            muls    d2,d0               * cosh破壊
134:            add.l   d0,d0
135:            add.l   d0,d0
136:            swap    d0
137:            move.w  d0,pers12+2         * pers12+2=cosh*cosp
138:
139:            movem.w cx(pc),d1-d3
140:            swap    d1
141:            swap    d2
142:            clr.w   d1
143:            clr.w   d2
144:            asr.l   #2,d1
145:            asr.l   #2,d2
146:            move.l  d1,a5
147:            move.l  d2,a6
148:            move.w  d3,pers13+2
149:
150:            move.w  minz(pc),a0
151:            move.l  point_adr(pc),a1
152:            move.l  obj_adr(pc),a2
153:            movem.w dx(pc),d5-d7
154: pers1:
155:            movem.w (a2)+,d1-d3
156:            sub.w   d5,d1
157:            sub.w   d6,d2
158:            sub.w   d7,d3
159: pers2:
160:            move.w  #0,d0
161:            muls    d1,d0
162: pers3:
163:            move.w  #0,d4
164:            muls    d2,d4
165:            add.l   d4,d0
166: pers4:
167:            move.w  #0,d4
168:            muls    d3,d4
169:            add.l   d4,d0
170: pers5:
171:            add.l   a5,d0
172:            move.l  d0,a3    * a3 … x2
173: *
174: pers6:
175:            move.w  #0,d0
176:            muls    d1,d0
177: pers7:
178:            move.w  #0,d4
179:            muls    d2,d4
180:            add.l   d4,d0
181: pers8:
182:            move.w  #0,d4
183:            muls    d3,d4
184:            add.l   d4,d0
185: pers9:
186:            add.l   a6,d0
187:            move.l  d0,a4    * a4 … y2
188: *
189: pers10:
190:            move.w  #0,d0
191:            muls    d1,d0
192: pers11:
193:            move.w  #0,d4
194:            muls    d2,d4
195:            add.l   d4,d0
196: pers12:
197:            move.w  #0,d4
198:            muls    d3,d4
199:            add.l   d4,d0
200:            asl.l   #2,d0
201:            swap    d0
202: pers13:
203:            add.w   #0,d0
204:
205:            move.l  a3,d1
206:            asr.l   #7,d1
207:            move.l  a4,d2
208:            asr.l   #7,d2
209:            movem.l d1-d2,6(a1)         * 回転移動後X，Y座標
210:            move.w  d0,14(a1)           * 回転移動後Z座標
211:            bmi     pers17              * Z＜0
212:            add.w   a0,d0               * add.w minz(pc),d0
213:            divs    d0,d1
214:            add.w   d1,d1
215: point_ox1:
216:            add.w   #0,d1
217:            divs    d0,d2
218:            add.w   d2,d2
219: point_oy1:
220:            add.w   #0,d2
221: pers17:
222:            movem.w d1-d2,(a1)          * 透視図X，Y座標
223:            lea.l   16(a1),a1           * 1ワード余分
224:            subq.w  #1,pct_work
225:            bne     pers1
226:            move.l  a1,point_adr
227: perspective_end:
228:            move.l  (sp)+,a0
229:            rts
230:
231: *          work area
232:
233:            .data
234: cx:
235:            dc.w    0
236: cy:
237:            dc.w    0
238: cz:
239:            dc.w    0
240: dx:
241:            dc.w    0
242: dy:
243:            dc.w    0
244: dz:
245:            dc.w    0
246: head:
247:            dc.w    0
248: pitch:
249:            dc.w    0
250: bank:
251:            dc.w    0
252: cosh
253:            dc.w    0
254: sinh:
255:            dc.w    0
256: cosp:
257:            dc.w    0
258: sinp:
259:            dc.w    0
260: cosb:
261:            dc.w    0
262: sinb:
263:            dc.w    0
264: pct:
265:            ds.w    1         * ページ0にある頂点の数
266: pct2:
267:            ds.w    1         * ページ1にある頂点の数
268: pct_work:
269:            ds.w    1         * 直前にデータセットした頂点の数
270: obj_adr:
271:            ds.l    1         * 線分データのアドレス
272: point_adr:
273:            ds.l    1         * 3D－2D変換データの格納ポインタ
274: line_adr:
275:            ds.l    1         * 線分データの格納ポインタ
276: data_adr:
277:            ds.l    1         * 物体構造体へのポインタ
278:
279:            .bss
280:
281: *-----------------------------------------------------
282:            * 座標1つが2ワード
283: disp_buf:
284:            ds.w    2*8192  * ページ0の2D変換した座標
285: disp_buf2:
286:            ds.w    2*8192  * ページ1の2D変換した座標
287:
288: *-----------------------------------------------------
289:            * ライン1本が2ワード
290: lin_buf:
291:            ds.w    2*8192  * ページ0の線分データ
292: lin_buf2:
293:            ds.w    2*8192  * ページ1の線分データ
294:
295: *-----------------------------------------------------
296:            * data_buf  +0L  2Dの座標データへのポインタ
297:            *           +4W  線分の数
298:            *           +6W  物体のカラーコード（拡張）
299:            * 以下繰り返し
300: data_buf:
301:            ds.b    1600    * ページ0の物体200個分の構造体
302: data_buf2:
303:            ds.b    1600    * ページ1の物体200個分の構造体
304:
305: *-----------------------------------------------------
306: p_end:
307:            .end
308:
```

リスト14

```
 1: *
 2: * disp_flame.s  version 1.40
 3: *
 4:            .xdef   disp_flame
 5:            .xdef   point_ox2
 6:            .xdef   point_oy2
 7:            .xdef   df_putch1
 8:            .xdef   df_putch2
 9:            .xdef   df_mode
10:            .xdef   df_mode2
11:
12:            .xref   line_main
13:
14:            .xref   lin_buf
15:            .xref   lin_buf2
16:            .xref   disp_buf
17:            .xref   disp_buf2
18:            .xref   data_buf
19:            .xref   data_buf2
20:
21:            .xref   pct
22:            .xref   pct2
23:            .xref   page
24:            .xref   color
25:            .xref   line_adr
26:            .xref   point_adr
27:            .xref   data_adr
28:
29:            .xref   l_gbase
30:            .xref   r_gbase
31:
32:            .xref   crt_mode
33:            .xref   obj_no
34:            .xref   obj_no2
35:            .xref   minz
36:            .xref   maxz
37:
38: disp_flame:
39:            move.l  a0,-(sp)
40:
41:            movem.l work_list(pc),d1-d6
42:            lea.l   pct(pc),a0
43:            lea.l   pct2(pc),a2
44:            lea.l   obj_no(pc),a4
45:            lea.l   obj_no2(pc),a6
46:            tst.b   page
47:            beq     df2
48:            exg.l   d1,d2
49:            exg.l   d3,d4
50:            exg.l   d5,d6
51:            exg.l   a0,a2
52:            exg.l   a4,a6
53: df2:
54:            clr.w   (a2)                * 裏画面のpctをクリア
55:
56:            move.l  d2,line_adr
57:            move.l  d4,data_adr
58:            move.l  d6,point_adr
```

▶大学でパーセプトロンの講義をやっているが全然わからない。　山下　智史(20)石川県

```
 59:         move.l  d1,d5           * ワークにコピー
 60:
 61:         eori.b  #1,page
 62:         bne     df_putch1
 63:         bclr.b  #3,l_gbase+3    * vram base address $C00000
 64:         bclr.b  #3,r_gbase+3    *
 65:         bra     df3
 66: df_putch1:
 67:         bset.b  #3,l_gbase+3    * vram base address $C80000
 68: df_putch2:
 69:         bset.b  #3,r_gbase+3    *
 70: df3:
 71:         movea.l d1,a0           * 線分バッファ
 72:         movea.l d3,a3           * ２Ｄ変換後のデータバッファ
 73:
 74:         move.w  (a4),d1         * 表示する物体の数
 75:         beq     disp_flame512   * 物体はない
 76:
 77:         movem.l d1-d5,-(sp)
 78: df3_1:
 79:         movea.l (a3)+,a5        * 格納アドレス
 80:         move.w  (a3)+,d7        * 線分の数
 81:         subq.w  #1,d7
 82: df_mode:
 83:         move.w  (a3)+,color     * パレットコード
 84:                                 * コンパチモードでは
 85:                                 *  NOP が書込まれる
 86: df4:
 87:         move.l  (a0)+,d5        * d5.l=頂点１| 頂点２
 88:         lsl.l   #4,d5           * 頂点番号を１６倍する
 89:         move.w  d5,d6           * d6.w = 頂点２
 90:         swap    d5              * d5.w = 頂点１
 91:         movem.w (a5,d5.w),d1-d2 * x1,y1
 92:         movem.w (a5,d6.w),d3-d4 * x2,y2
 93:
 94:         tst.w   14(a5,d5.w)     * 頂点１のＺ座標
 95:         bpl     df5
 96:         tst.w   14(a5,d6.w)     * 頂点２のＺ座標
 97:         bmi     df11            * 不可視（投影面より手前）
 98:         jsr     clip_z(pc)      * z1<0, z2>=0
 99:         movem.w clip_dat(pc),d1-d2
100:         bra     df6
101: df5:
102:         tst.w   14(a5,d6.w)
103:         bpl     df6             * 可視
104:         jsr     clip_z(pc)      * z2<0, z1>=0
105:         movem.w clip_dat(pc),d3-d4
106: df6:
107:         move.w  14(a5,d5.w),d0
108:         cmp.w   maxz(pc),d0
109:         ble     df7
110:         move.w  14(a5,d6.w),d0
111:         cmp.w   maxz(pc),d0
112:         bgt     df11            * 不可視（後方面より奥）
113:         jsr     clip_z2(pc)     * z1>z_max, z2<=z_max
114:         movem.w clip_dat(pc),d1-d2
115:         bra     df8
116: df7:
117:         move.w  14(a5,d6.w),d0
118:         cmp.w   maxz(pc),d0
119:         ble     df8             * 可視
120:         jsr     clip_z2(pc)     * z2>z_max, z1<=z_max
121:         movem.w clip_dat(pc),d3-d4
122: df8:
123:         move.w  d7,-(sp)
124:         jsr     line_main(pc)
125:         move.w  (sp)+,d7
126: df11:
127:         dbf     d7,df4          * 物体を１つ表示するまでループ
128:         subq.w  #1,(a4)         * すべての物体を表示するまでループ
129:         bne     df3_1
130:         movem.l (sp)+,d1-d5
131:         move.w  d1,(a4)
132:
133:         move.w  crt_mode(pc),d6
134:         cmpi.w  #16,d6
135:         bne     disp_flame512   * 512*512だ
136:         movea.l d5,a0
137:         movea.l d3,a3
138:         move.w  d1,(a6)
139:         bra     disp_flame768   * 768*512だ
140: disp_flame512:
141:         move.b  $e82601,d5
142:         subq.w  #8,d6
143:         bcc     df13
144:                                 * ページが４枚の時
145:         andi.b  #%1111_1100,d5
146:         moveq.l #%0001,d6
147:         tst.b   page
148:         beq     df14
149:         moveq.l #%0010,d6
150:         bra     df14
151: df13:
152:                                 * ページが２枚の時
153:         andi.b  #%1111_0000,d5
154:         moveq.l #%0011,d6
155:         tst.b   page
156:         beq     df14
157:         moveq.l #%1100,d6
158: df14:
159:         or.b    d6,d5
160:         move.b  d5,$e82601      * 表示ページ切り替え
161:
162:         tst.b   page
163:         beq     df18            * 表示ページが０ならアクセスページは１
164:         bclr.b  #3,l_gbase+3    * vram base address $C00000
165:         bclr.b  #3,r_gbase+3    *
166:         bra     df19
167: df18:
168:         bset.b  #3,l_gbase+3    * vram base address $C80000
169:         bset.b  #3,r_gbase+3    *
170: df19:
171:         movea.l d2,a0           * 線分バッファ
172:         movea.l d4,a3           * ２Ｄ変換後のデータバッファ
173: disp_flame768:
174:         tst.w   (a6)
175:         beq     disp_flame_end
176:
177:         move.w  color(pc),-(sp)
178:         clr.w   color
179: df20_1:
180:         movea.l (a3)+,a5        * 格納アドレス
181:         move.w  (a3)+,d7        * 線分の数
182:         subq.w  #1,d7
183: df_mode2:
184:         addq.w  #2,a0           * パレットコード無視
185:                                 * コンパチモードでは
186:                                 *  NOP が書込まれる
187: df20:
188:         move.l  (a0)+,d5        * d5.l=頂点１| 頂点２
189:         lsl.l   #4,d5           * 頂点番号を１６倍する
190:         move.w  d5,d6           * d6.w = 頂点２
191:         swap    d5              * d5.w = 頂点１
192:
193:         movem.w (a5,d5.w),d1-d2 * x1,y1
194:         movem.w (a5,d6.w),d3-d4 * x2,y2
195:
196:         tst.w   14(a5,d5.w)     * 頂点１のＺ座標
197:         bpl     df25
198:         tst.w   14(a5,d6.w)     * 頂点２のＺ座標
199:         bmi     df29            * 不可視（投影面より手前）
200:         jsr     clip_z(pc)      * z1<0, z2>=0
201:         movem.w clip_dat(pc),d1-d2
202:         bra     df26
203: df25:
204:         tst.w   14(a5,d6.w)
205:         bpl     df26            * 可視
206:         jsr     clip_z(pc)      * z2<0, z1>=0
207:         movem.w clip_dat(pc),d3-d4
208: df26:
209:         move.w  14(a5,d5.w),d0
210:         cmp.w   maxz(pc),d0
211:         ble     df27
212:         move.w  14(a5,d6.w),d0
213:         cmp.w   maxz(pc),d0
214:         bgt     df29            * 不可視（後方面より奥）
215:         jsr     clip_z2(pc)     * z1>z_max, z2<=z_max
216:         movem.w clip_dat(pc),d1-d2
217:         bra     df28
218: df27:
219:         move.w  14(a5,d6.w),d0
220:         cmp.w   maxz(pc),d0
221:         ble     df28            * 可視
222:         jsr     clip_z2(pc)     * z2>z_max, z1<=z_max
223:         movem.w clip_dat(pc),d3-d4
224: df28:
225:         move.w  d7,a4
226:         jsr     line_main(pc)
227:         move.w  a4,d7
228: df29:
229:         dbf     d7,df20         * 物体を１つ表示するまでループ
230:         subq.w  #1,(a6)
231:         bne     df20_1
232:         move.w  (sp)+,color
233: disp_flame_end:
234:         clr.w   (a6)            * 裏画面のｌｃｔをクリア
235:         move.l  (sp)+,a0
236:         rts
237:
238:           Ｚ軸クリッピング処理
239:
240: clip_z:
241:         movem.l d0-d7/a0-a6,-(sp)
242:         jsr     clip_main(pc)
243:         movem.l (sp)+,d0-d7/a0-a6
244:         rts
245: clip_z2:
246:         movem.l d0-d7/a0-a6,-(sp)
247:         jsr     clip_main2(pc)
248:         movem.l (sp)+,d0-d7/a0-a6
249:         rts
250:
251: clip_main:
252:         move.l  6(a5,d5.w),d0   *  Ｘ１
253:         move.l  6(a5,d6.w),d1   *  Ｘ２
254:         move.l  10(a5,d5.w),d2  *  Ｙ１
255:         move.l  10(a5,d6.w),d3  *  Ｙ２
256:         move.w  14(a5,d5.w),d4  *  Ｚ１
257:         move.w  14(a5,d6.w),d7  *  Ｚ２
258:         cmp.w   d4,d7
259:         bgt     cl_2
260:         exg     d0,d1
261:         exg     d2,d3
262:         exg     d4,d7
263: cl_2:                           * z1≦z2が保証される
264:         move.l  d1,a3
265:         move.l  d3,a4
266:         move.w  d7,a5
267:         add.l   d0,d1           * d2.l=x2+x1
268:         add.l   d2,d3           * d4.l=y2+y1
269:         add.w   d4,d7           * d7.w=z2+z1
270:         asr.l   #1,d1
271:         asr.l   #1,d3
272:         asr.w   #1,d7
273:         tst.w   d7
274:         beq     clip_end
275:         cmpi.w  #1,d7
276:         beq     clip_end
277:         bgt     cl_2
278:         cmpi.w  #-1,d7
279:         beq     clip_end
280:         move.l  d1,d0
281:         move.l  d3,d2
282:         move.w  d7,d4
283:         move.l  a3,d1
284:         move.l  a4,d3
285:         move.w  a5,d7
286:         bra     cl_2
287:
288: clip_main2:
289:         move.l  6(a5,d5.w),d0   *  Ｘ１
290:         move.l  6(a5,d6.w),d1   *  Ｘ２
291:         move.l  10(a5,d5.w),d2  *  Ｙ１
292:         move.l  10(a5,d6.w),d3  *  Ｙ２
293:         move.w  14(a5,d5.w),d4  *  Ｚ１
294:         move.w  14(a5,d6.w),d7  *  Ｚ２
295:         move.w  maxz(pc),d5
296:         move.w  d5,d6
297:         addq.w  #1,d5   * d5.w = maxz+1
298:         subq.w  #1,d6   * d6.w = maxz-1
299:         cmp.w   d4,d7
300:         bgt     cl_m2
301:         exg     d0,d1
302:         exg     d2,d3
303:         exg     d4,d7
304: cl_m2:                  * z1≦z2が保証される
305:         move.l  d1,a3
306:         move.l  d3,a4
307:         move.w  d7,a5
308:         add.l   d0,d1           * d2.l=x2+x1
309:         add.l   d2,d3           * d4.l=y2+y1
310:         add.w   d4,d7           * d7.w=z2+z1
311:         asr.l   #1,d1
312:         asr.l   #1,d3
313:         asr.w   #1,d7
314:         cmp.w   maxz(pc),d7
315:         beq     clip_end2
316:         cmp.w   d5,d7
317:         beq     clip_end2
318:         bgt     cl_m2
319:         cmp.w   d6,d7
320:         beq     clip_end2
321:         move.l  d1,d0
322:         move.l  d3,d2
323:         move.w  d7,d4
324:         move.l  a3,d1
325:         move.l  a4,d3
326:         move.w  a5,d7
327:         bra     cl_m2
328:
329: clip_end:
330:         move.w  minz(pc),d7
331:         bra     clip_sk1
332: clip_end2:
333:         move.w  maxz(pc),d7
334:         add.w   minz(pc),d7
335: clip_sk1:
336:         divs    d7,d1
```

▶ IO-735Xがほしくなりました。しかし,大きすぎて家の机には乗せることができないでしょう（悲）。
綾野　雅之(17)愛知県

```
337:         add.w   d1,d1
338: point_ox2:
339:         add.w   #0,d1
340:         divs    d7,d3
341:         add.w   d3,d3
342: point_oy2:
343:         add.w   #0,d3
344:         lea.l   clip_dat(pc),a1
345:         move.w  d1,0(a1)          * 透視図X座標
346:         move.w  d3,2(a1)          * 透視図Y座標
347:         rts
348:
349:         .data
350:
351: work_list:
352:         dc.l    lin_buf        * d1.l
353:         dc.l    lin_buf2       * d2.l
354:         dc.l    data_buf       * d3.l
355:         dc.l    data_buf2      * d4.l
356:         dc.l    disp_buf       * d5.l
357:         dc.l    disp_buf2      * d6.l
358: clip_dat:
359:         ds.w    2
360:
361:         .end
362:
```

リスト 15

```
  1: *
  2: * scrmod.s      version 1.11
  3: *
  4:         .include        iocscall.mac
  5:
  6:         .xdef   scrmod
  7:         .xdef   crt_mode
  8:
  9:         .xref   apage_main
 10:
 11:         .xref   point_ox1
 12:         .xref   point_oy1
 13:         .xref   point_ox2
 14:         .xref   point_oy2
 15:         .xref   b_sft_time
 16:         .xref   c_sft_time
 17:         .xref   f_sft_time
 18:         .xref   l_sft_time
 19:         .xref   r_sft_time
 20:         .xref   r_sft_time2
 21:
 22:         .xref   b_gbase
 23:         .xref   c_gbase
 24:         .xref   f_gbase
 25:         .xref   l_gbase
 26:         .xref   r_gbase
 27:         .xref   r_gbase2
 28:
 29:         .xref   b_pixel
 30:         .xref   c_pixel
 31:         .xref   f_pixel
 32:         .xref   l_pixel
 33:
 34:         .xref   df_putch1
 35:         .xref   df_putch2
 36:         .xref   df_mode
 37:         .xref   df_mode2
 38:         .xref   data_mode
 39:         .xref   wipe_putch
 40:
 41:         .xref   page
 42:         .xref   minx
 43:         .xref   miny
 44:         .xref   maxx
 45:         .xref   maxy
 46:
 47:         .text
 48:         .even
 49:
 50: scrmod:
 51:         move.w  (a0)+,d2
 52:         move.w  #$4a79,df_mode  * disp_flame コンパチ
 53:         move.w  #$4e71,df_mode2 * disp_flame コンパチ
 54:         move.w  #$4e71,data_mode
 55:         cmpi.w  #$100,d2
 56:         bcs     scrmod1
 57:         subi.w  #$100,d2
 58:         move.w  #$33db,df_mode  * disp_flame 拡張(カラー対応)
 59:         move.w  #$544b,df_mode2 * disp_flame 拡張(カラー対応)
 60:         move.w  #$34d8,data_mode
 61: scrmod1:
 62:         cmpi.w  #2,d2
 63:         beq     skip            * 実画面が1024*1024だ
 64:
 65: *       実画面512*512の時
 66:
 67:         move.b  #10,b_sft_time+1
 68:         move.b  #10,c_sft_time+1
 69:         move.b  #10,f_sft_time+1
 70:         move.b  #10,l_sft_time+1
 71:         move.b  #10,r_sft_time+1
 72:         move.b  #10,r_sft_time2+1
 73:         move.w  #1024,b_pixel+2
 74:         move.w  #1024,c_pixel+2
 75:         move.w  #1024,f_pixel+2
 76:         move.w  #1024,l_pixel+2
 77:         move.w  #$08f9,df_putch1        * bset
 78:         move.w  #$08f9,df_putch2
 79:         move.w  #$08f9,wipe_putch
 80:         bra     scrmod2
 81: skip:
 82:
 83: *       実画面1024*1024の時
 84:
 85:         move.b  #11,b_sft_time+1
 86:         move.b  #11,c_sft_time+1
 87:         move.b  #11,f_sft_time+1
 88:         move.b  #11,l_sft_time+1
 89:         move.b  #11,r_sft_time+1
 90:         move.b  #11,r_sft_time2+1
 91:         move.w  #2048,b_pixel+2
 92:         move.w  #2048,c_pixel+2
 93:         move.w  #2048,f_pixel+2
 94:         move.w  #2048,l_pixel+2
 95:         move.w  #$08b9,df_putch1        * bclr
 96:         move.w  #$08b9,df_putch2
 97: scrmod2:
 98:         add.w   d2,d2
 99:         move.w  scrmod_data(pc,d2.w),d1
100:         move.w  d1,crt_mode
101:         IOCS    _CRTMOD
102:         IOCS    _G_CLR_ON
103:         add.w   d2,d2
104:         add.w   d2,d2
105:         movem.l window_data(pc,d2.w),d1-d2
106:         movem.l d1-d2,minx
107:         clr.w   d1
108:         bsr     apage_main      * ページ0
109:         moveq.l #1,d1
110:         IOCS    _VPAGE
111:
112:         clr.b   page
113:
114:         move.w  minx(pc),d1
115:         add.w   maxx(pc),d1
116:         asr.w   #1,d1
117:         move.w  d1,point_ox1+2
118:         move.w  d1,point_ox2+2
119:
120:         move.w  miny(pc),d1
121:         add.w   maxy(pc),d1
122:         asr.w   #1,d1
123:         move.w  d1,point_oy1+2
124:         move.w  d1,point_oy2+2
125:         rts
126:
127: scrmod_data:
128:         dc.w    10      * 256*256 / 256色 2面
129:         dc.w    8       * 512*512 / 256色 2面
130:         dc.w    16      * 768*512 /  16色 1面
131:         dc.w    6       * 256*256 /  16色 4面
132:         dc.w    4       * 512*512 /  16色 4面
133:
134: window_data:
135:         dc.w    0,0,255,255
136:         dc.w    0,0,511,511
137:         dc.w    0,0,767,511
138:         dc.w    0,0,255,255
139:         dc.w    0,0,511,511
140:
141:         .data
142:
143: crt_mode:               * 画面モード保存用ワーク
144:         ds.w    1
145:
146:         .end
```

リスト 16

```
  1: *
  2: * set_color.s
  3: *
  4:         .xdef   set_color
  5:         .xdef   color
  6:         .xdef   page
  7:
  8:         .text
  9:         .even
 10:
 11: set_color:
 12:         move.w  (a0)+,color
 13:         rts
 14:
 15:         .data
 16:
 17: color:
 18:         ds.w    1
 19: page:
 20:         ds.w    1
 21:
 22:         .end
 23:
```

リスト 17

```
  1: *
  2: * magic initialize      version 1.11
  3: *
  4:         .include        iocscall.mac
  5:
  6:         .xdef   init
  7:
  8:         .xref   lin_buf
  9:         .xref   lin_buf2
 10:         .xref   disp_buf
 11:         .xref   disp_buf2
 12:         .xref   data_buf
 13:         .xref   data_buf2
 14:
 15:         .xref   line_adr
 16:         .xref   point_adr
 17:         .xref   data_adr
 18:
 19:         .xref   page
 20:         .xref   pct
 21:         .xref   pct2
 22:         .xref   pct_work
 23:         .xref   obj_no
 24:         .xref   obj_no2
 25:         .xref   crt_mode
 26:         .xref   disp_mode
 27:         .xref   minz
 28:         .xref   maxz
 29:
 30:         .text
 31:         .even
 32:
 33: init:
 34:         move.l  #disp_buf,point_adr
 35:         move.l  #lin_buf,line_adr
 36:         move.l  #data_buf,data_adr
 37:         move.w  #50,minz
 38:         move.w  #2000,maxz
 39:         moveq.l #-1,d1
 40:         IOCS    _CRTMOD
 41:         move.w  d0,crt_mode
 42:         move.w  #2,disp_mode    * or mode
 43:
 44:         lea.l   disp_buf,a1
 45:         lea.l   disp_buf+1,a2
 46:         move.l  #32768-1,d2
 47:         move.b  #%00_00_00_01,d1
 48:         clr.l   (a1)
 49:         IOCS    _DMAMOVE
 50:         lea.l   disp_buf2,a1
 51:         lea.l   disp_buf2+1,a2
 52:         clr.l   (a1)
 53:         IOCS    _DMAMOVE
 54:         lea.l   lin_buf,a1
```

▶たんぼに水が入ってめちゃ寒い。わかるんだろうな都会人には。　橘　正彦(18)福島県

```
55:         lea.l   lin_buf+1,a2
56:         clr.l   (a1)
57:         IOCS    _DMAMOVE
58:         lea.l   lin_buf2,a1
59:         lea.l   lin_buf2+1,a2
60:         clr.l   (a1)
61:         IOCS    _DMAMOVE
62:         lea.l   data_buf,a1
63:         lea.l   data_buf+1,a2
64:         move.l  #1600-1,d2
65:         clr.l   (a1)
66:         IOCS    _DMAMOVE
67:         lea.l   data_buf2,a1
68:         lea.l   data_buf2+1,a2
69:         clr.l   (a1)
70:         IOCS    _DMAMOVE
71:
72:         clr.b   page
73:         clr.w   pct
74:         clr.w   pct2
75:         clr.w   pct_work
76:         clr.w   obj_no
77:         clr.w   obj_no2
78:         rts
79:
80:         .end
81:
```

リスト 18

```
 1: *
 2: * sin102.s       version 1.02
 3: *
 4:         .xdef   sinhl
 5:         .xdef   coshl
 6:
 7:         .text
 8:         .even
 9:
10: coshl:
11:         addi.w  #90,d1          * 8 サインテーブルを使えるように角度を変換する
12: sinhl:
13:         subi.w  #360,d1         * 8
14:         bpl     sinhl           * 10
15: sinhl2:
16:         addi.w  #360,d1         * 8
17:         bmi     sinhl2          * 10
18:
19:         move.w  #180,d3         * 8
20:         sub.w   d3,d1           * 4
21:         bpl     sinhl3          * 10
22:         add.w   d3,d1           * 4
23: sinhl3:
24:         scs.b   d7              * 4 or 6
25:         cmpi.w  #90,d1          * 8
26:         bmi     sinhl4          * 10
27:         sub.w   d1,d3           * 4
28:         move.w  d3,d1           * 4
29: sinhl4:
30:         add.w   d1,d1           * 4
31:         move.w  sin_table(pc,d1.w),d1   * 14
32:         tst.b   d7              * 4
33:         bne     sinhl5          * 10
34:         neg.w   d1              * 4
35: sinhl5:
36:         move.w  d1,(a1)+
37:         rts                     * 16
38:
39:                                 * 152 or 154(clock)
40:
41: sin_table:
42:         dc.w    $0000,$011d,$023b,$0359
43:         dc.w    $0476,$0593,$06b0,$07cc
44:         dc.w    $08e8,$0a03,$0b1d,$0c36
45:         dc.w    $0d4e,$0e65,$0f7b,$1090
46:         dc.w    $11a4,$12b6,$13c6,$14d6
47:
48:         dc.w    $15e3,$16ef,$17f9,$1901
49:         dc.w    $1a07,$1b0c,$1c0e,$1d0e
50:         dc.w    $1e0b,$1f07,$2000,$20f6
51:         dc.w    $21ea,$22db,$23c9,$24b5
52:         dc.w    $259e,$2684,$2766,$2846
53:
54:         dc.w    $2923,$29fc,$2ad3,$2ba5
55:         dc.w    $2c75,$2d41,$2e09,$2ece
56:         dc.w    $2f8f,$304d,$3106,$31bc
57:         dc.w    $326e,$331c,$33c6,$346c
58:         dc.w    $350e,$35ac,$3646,$36db
59:
60:         dc.w    $376c,$37f9,$3882,$3906
61:         dc.w    $3985,$3a00,$3a77,$3ae9
62:         dc.w    $3b56,$3bbf,$3c23,$3c83
63:         dc.w    $3cde,$3d34,$3d85,$3dd1
64:         dc.w    $3e19,$3e5c,$3e99,$3ed2
65:
66:         dc.w    $3f07,$3f36,$3f60,$3f85
67:         dc.w    $3fa6,$3fc1,$3fd8,$3fe9
68:         dc.w    $3ff6,$3ffd,$4000
69:
70:         .end
71:
72:
```

リスト 19

```
  1: *
  2: * raster.s
  3: *
  4:         .xdef   raster
  5:         .xdef   raster2
  6:         .xdef   raster_pset
  7:         .xdef   raster_main
  8:         .xdef   raster_main2
  9:         .xdef   ras_mode1
 10:         .xdef   ras_mode2
 11:         .xdef   r_gbase
 12:         .xdef   r_gbase2
 13:         .xdef   r_sft_time
 14:         .xdef   r_sft_time2
 15:         .xdef   r_dot
 16:
 17:         .xref   disp_mode
 18:         .xref   color
 19:         .xref   minx
 20:         .xref   miny
 21:         .xref   maxx
 22:         .xref   maxy
 23:
 24:         .text
 25:         .even
 26:
 27: raster:
 28:         cmpi.w  #2,disp_mode
 29:         bne     raster2         * 描画モードに対応させる
 30: raster_pset:
 31:         movem.l d1-d7/a2,-(sp)
 32: r_gbase:
 33:         lea.l   $c00000+2,a1
 34:         cmp.w   d1,d2
 35:         bge     raster_sk1
 36:         exg.l   d1,d2
 37: raster_sk1:
 38:         cmp.w   maxy(pc),d3
 39:         bgt     raster_rts
 40:         cmp.w   miny(pc),d3
 41:         blt     raster_rts
 42:
 43:         cmp.w   maxx(pc),d2
 44:         ble     raster_sk2
 45:         cmp.w   maxx(pc),d1
 46:         bgt     raster_rts
 47:         move.w  maxx(pc),d2
 48: raster_sk2:
 49:         cmp.w   minx(pc),d1
 50:         bge     raster_sk3
 51:         cmp.w   minx(pc),d2
 52:         blt     raster_rts
 53:         move.w  minx(pc),d1
 54: raster_sk3:
 55:         sub.w   d2,d1
 56:         neg.w   d1
 57:         addq.w  #1,d1   * 0でも1ドットある
 58: r_sft_time:
 59:         moveq.l #10,d0
 60:         asl.l   d0,d3   * d3を1024、または2048倍する
 61:         add.w   d2,d2
 62:         add.w   d2,d3
 63:         adda.l  d3,a1   * vram address
 64:         jsr     raster_main2(pc)
 65: raster_rts:
 66:         movem.l (sp)+,d1-d7/a2
 67: raster_end:
 68:         rts
 69:
 70: raster_main:
 71:         movem.l d3-d7,-(sp)
 72:         jsr     raster_main2
 73:         movem.l (sp)+,d3-d7
 74:         rts
 75:
 76: raster_main2:
 77:         move.w  color(pc),d0
 78:         move.w  d1,d2
 79:         andi.w  #15,d1
 80:         neg.w   d1
 81:         add.w   #16,d1
 82:         add.w   d1,d1
 83:         jmp     raster_draw1(pc,d1.w)
 84: raster_draw1:
 85:         move.w  d0,-(a1)
 86:         move.w  d0,-(a1)
 87:         move.w  d0,-(a1)
 88:         move.w  d0,-(a1)
 89:         move.w  d0,-(a1)
 90:         move.w  d0,-(a1)
 91:         move.w  d0,-(a1)
 92:         move.w  d0,-(a1)
 93:         move.w  d0,-(a1)
 94:         move.w  d0,-(a1)
 95:         move.w  d0,-(a1)
 96:         move.w  d0,-(a1)
 97:         move.w  d0,-(a1)
 98:         move.w  d0,-(a1)
 99:         move.w  d0,-(a1)
100:         move.w  d0,-(a1)
101:         lsr.w   #4,d2
102:         beq     raster_end
103: long_raster:
104:         lea.l   raster_draw2(pc),a2
105:         add.w   d2,d2
106:         add.w   d2,d2
107:         suba.w  d2,a2
108:
109:         move.w  d0,d1
110:         swap    d0
111:         move.w  d1,d0
112:         move.l  d0,d1
113:         move.l  d0,d2
114:         move.l  d0,d3
115:         move.l  d0,d4
116:         move.l  d0,d5
117:         move.l  d0,d6
118:         move.l  d0,d7
119:         jmp     (a2)
120:
121:         movem.l d0-d7,-(a1)
122:         movem.l d0-d7,-(a1)
123:         movem.l d0-d7,-(a1)
124:         movem.l d0-d7,-(a1)
125:         movem.l d0-d7,-(a1)
126:         movem.l d0-d7,-(a1)
127:         movem.l d0-d7,-(a1)
128:         movem.l d0-d7,-(a1)
129:         movem.l d0-d7,-(a1)
130:         movem.l d0-d7,-(a1)
131:         movem.l d0-d7,-(a1)
132:         movem.l d0-d7,-(a1)
133:         movem.l d0-d7,-(a1)
134:         movem.l d0-d7,-(a1)
135:         movem.l d0-d7,-(a1)
136:         movem.l d0-d7,-(a1)
137:         movem.l d0-d7,-(a1)
138:         movem.l d0-d7,-(a1)
139:         movem.l d0-d7,-(a1)
140:         movem.l d0-d7,-(a1)
141:         movem.l d0-d7,-(a1)
142:         movem.l d0-d7,-(a1)
143:         movem.l d0-d7,-(a1)
144:         movem.l d0-d7,-(a1)
145:         movem.l d0-d7,-(a1)
146:         movem.l d0-d7,-(a1)
147:         movem.l d0-d7,-(a1)
148:         movem.l d0-d7,-(a1)
149:         movem.l d0-d7,-(a1)
150:         movem.l d0-d7,-(a1)
151:         movem.l d0-d7,-(a1)
152:         movem.l d0-d7,-(a1)
153:         movem.l d0-d7,-(a1)
154:         movem.l d0-d7,-(a1)
```

▶横スクロールゲームは左から右に進んでいくものばかり。右から左へスクロールするのも面白いと思う。皆,保守的なのね。
山野井　直樹(23)京都府

```
155:          movem.l d0-d7,-(a1)
156:          movem.l d0-d7,-(a1)
157:          movem.l d0-d7,-(a1)
158:          movem.l d0-d7,-(a1)
159:          movem.l d0-d7,-(a1)
160:          movem.l d0-d7,-(a1)
161:          movem.l d0-d7,-(a1)
162:          movem.l d0-d7,-(a1)
163:          movem.l d0-d7,-(a1)
164:          movem.l d0-d7,-(a1)
165:          movem.l d0-d7,-(a1)
166:          movem.l d0-d7,-(a1)
167:          movem.l d0-d7,-(a1)
168:          movem.l d0-d7,-(a1)
169:          movem.l d0-d7,-(a1)
170:          movem.l d0-d7,-(a1)
171:          movem.l d0-d7,-(a1)
172:          movem.l d0-d7,-(a1)
173:          movem.l d0-d7,-(a1)
174:          movem.l d0-d7,-(a1)
175:          movem.l d0-d7,-(a1)
176:          movem.l d0-d7,-(a1)
177:          movem.l d0-d7,-(a1)
178:          movem.l d0-d7,-(a1)
179:          movem.l d0-d7,-(a1)
180:          movem.l d0-d7,-(a1)
181:          movem.l d0-d7,-(a1)
182:          movem.l d0-d7,-(a1)
183:          movem.l d0-d7,-(a1)
184: raster_draw2:
185:          rts
186:
187: *        描画モード対応ルーチン
188:
189: raster2:
190:          movem.l d1-d3/a2,-(sp)
191: r_gbase2:
192:          lea.l   $c00000+2,a2    * プリデクリメント用に +2 する
193:          cmp.w   d1,d2
194:          bge     sk1
195:          exg.l   d1,d2
196: sk1:
197:          cmp.w   maxy(pc),d3
198:          bgt     raster_rts2
199:          cmp.w   miny(pc),d3
200:          blt     raster_rts2
201:
202:          cmp.w   maxx(pc),d2
203:          ble     sk2
204:          cmp.w   maxx(pc),d1
205:          bgt     raster_rts2
206:          move.w  maxx(pc),d2
207: sk2:
208:          cmp.w   minx(pc),d1
209:          bge     r_sft_time2
210:          cmp.w   minx(pc),d2
211:          blt     raster_rts2
212:          move.w  minx(pc),d1
213: r_sft_time2:
214:          moveq.l #10,d0  * 512*512 なら moveq.l #10,d0
215:          asl.l   d0,d3   * d3を1024、または2048倍する
216:          add.w   d2,d3
217:          add.w   d2,d3
218:          adda.l  d3,a2   * vram address
219:          sub.w   d2,d1
220:          neg.w   d1      * ピクセル数
221: sk3:
222:          jsr     raster_mode(pc)
223: raster_rts2:
224:          movem.l (sp)+,d1-d3/a2
225:          rts
226:
227: raster_mode:
228:          move.w  color(pc),d0
229:          move.w  d0,d2
230:          swap    d0
231:          move.w  d2,d0   * d0.l = color|color
232:
233:          tst.w   d1
234:          beq     r_dot
235:          lsr.w   #1,d1
236:          bcs     ras_mode2
237:          subq.w  #1,d1
238: ras_mode1:
239:          move.w  d0,-(a2)
240: ras_mode2:
241:          move.l  d0,-(a2)
242:          dbf     d1,ras_mode2
243: ras_main2_rts:
244:          rts
245: r_dot:
246:          move.w  d0,-(a2)
247:          rts
248:
249:          .end
```

リスト 20

```
 1: *
 2: * apage.s
 3: *
 4:          .xdef   apage
 5:          .xdef   apage_main
 6:
 7:          .xref   b_gbase
 8:          .xref   c_gbase
 9:          .xref   f_gbase
10:          .xref   r_gbase
11:          .xref   r_gbase2
12:          .xref   l_gbase
13:
14:          .text
15:          .even
16:
17: apage:
18:          move.w  (a0)+,d1
19: apage_main:
20:          andi.b  #3,d1            * ページ数は0～3
21:          lsl.w   #3,d1
22:          add.b   #$c0,d1
23:
24:          move.b  d1,b_gbase+3
25:          move.b  d1,c_gbase+3
26:          move.b  d1,f_gbase+3
27:          move.b  d1,r_gbase+3
28:          move.b  d1,r_gbase2+3
29:          move.b  d1,l_gbase+3
30:          rts
31:
32:          .end
```

リスト 21

```
 1: *
 2: * depth.s
 3: *
 4:          .xdef   depth
 5:          .xdef   minz
 6:          .xdef   maxz
 7:
 8:          .text
 9:          .even
10:
11: depth:
12:          move.l  (a0)+,minz
13:          rts
14:
15:          .data
16:
17: minz:
18:          dc.w    50
19: maxz:
20:          dc.w    2000
21:
22:          .end
```

リスト 22

```
as magic
as line
as spline
as box
as triangle
as boxfull
as circle
as window
as mode
as para
as wipe
as data
as perspective
as disp_flame
as scrmod
as init
as apage
as depth
as sincos
as raster
lk magic line spline box triangle boxfull circle window mode wipe para data pers
pective disp_flame scrmod set_color init sincos raster apage depth -x -oMAGIC.X
```

リスト 23

```
0000  1D 46 2D 6C 68 31 2D CE  : 90
0008  11 00 00 34 1C 00 00 31  : 92
0010  D1 F1 16 20 00 07 4D 41  : 8D
0018  47 49 43 2E 58 06 9B EA  : E4
0020  78 71 92 00 0C 6B F6 1D  : 05
0028  FD 1C 25 FB B1 C2 17 5B  : 1E
0030  E6 03 23 01 E0 92 21 9A  : 3A
0038  50 04 5D B8 02 A8 B2 CB  : 90
0040  34 D5 C0 7A EE CF 37 A7  : DE
0048  D5 E7 ED A5 E8 46 84 35  : 35
0050  EB 33 81 34 5D A0 1C 57  : 43
0058  3B 8C 85 20 5A 20 5F 3A  : 7F
0060  4E 73 73 09 5E D9 6D 3F  : 20
0068  32 18 F6 C7 F1 01 66 FB  : 5A
0070  0D 7A CC 19 E9 DF 94 4C  : 14
0078  69 C9 06 32 EF C4 60 12  : 8F
-------------------------------------
SUM:  16 5D AB 30 2F F7 F2 0C  D8EB

0080  4D 6A 64 5C 54 A5 7E 77  : 65
0088  3B 70 87 AF B2 B8 3F 34  : BE
0090  D2 62 B5 5B CE 47 72 3D  : 08
0098  E3 70 82 D7 A5 D0 88 FD  : A6
00A0  38 F4 52 B3 A0 AD 8C 95  : 9F
00A8  54 85 8F 86 78 9D 35 24  : 5C
00B0  0F FF A6 10 17 34 3F 13  : 61
00B8  E9 12 7B 21 EE F8 A9 09  : 2F
00C0  BF 04 11 B8 A0 CA 23 06  : 1F
00C8  44 14 36 68 94 85 AB 16  : D0
00D0  D2 C2 A6 78 E8 FB E3 A2  : 1A
00D8  62 37 49 D3 6A 9F E0 3E  : DC
00E0  35 67 FF 99 84 2F 42 BD  : E6
00E8  FE 4F 91 1A 6F EA FC 7C  : C9
00F0  EE 5B AE 92 E0 10 53 2A  : F6
00F8  DB 07 99 BF D7 76 C9 06  : 56
-------------------------------------
SUM:  F4 5F 31 16 C6 72 4B 1F  939B

0100  AA 8F 80 10 88 38 2E E7  : 9E
0108  47 CB 50 4B 09 8D 8A FF  : CC
0110  C1 69 36 63 E7 DC 06 41  : CD
0118  C3 B6 67 3C 5A FB CE F9  : 38
0120  66 E5 B2 DE 04 AA 9E 22  : 49
0128  49 F7 47 2F 43 EB 76 98  : F2
0130  94 31 0D 83 BE 52 BA 11  : 30
0138  16 8C 3A BF B5 BD BC 40  : 09
0140  E4 40 5A B1 00 97 4E 72  : 86
0148  04 94 40 8F 7A 34 03 CA  : E2
0150  D0 6F CC E9 A0 A8 AB 8A  : 71
0158  26 6E 75 03 3E 08 07 B8  : 11
0160  ED D1 31 7C E7 ED 45 F4  : 78
0168  03 12 6F DA 1B F7 DF 0E  : 5D
0170  98 6F A6 1B 5F A0 BF FD  : 83
0178  D9 79 74 27 2A 40 A0 21  : 18
-------------------------------------
SUM:  0D 8E 42 0D 6F 7F 9C C9  1285

0180  7C 1B CD 5B 7D 81 6B 08  : 30
0188  41 66 B7 7E 58 6E B5 DF  : 36
0190  37 98 C9 E2 CA 00 C4 40  : 48
0198  CE AD E8 0E 84 62 CC 3A  : 5D
01A0  94 DB 18 41 52 2B 41 6E  : F4
01A8  0D D5 F6 8E D7 6F A7 A1  : F4
01B0  D1 4F 55 6D 4D B1 DC 54  : 10
01B8  5D D9 F8 C7 25 8A DB C8  : 47
01C0  2B 35 E8 0B 3A DD 68 0B  : DD
01C8  F0 DF 44 B5 AF ED D9 FA  : 37
01D0  19 4B 34 EF 2D D3 DA 23  : 84
01D8  73 14 01 6B 02 B6 87 23  : 55
01E0  8D DA 59 9C 3F 4C 37 B7  : D5
01E8  16 BD 1D B8 F2 B9 71 5E  : 22
01F0  ED EC CE 28 0B B3 84 6A  : 7B
01F8  D4 54 07 5E 74 15 01 77  : 8E
-------------------------------------
SUM:  9C E8 3C C0 86 46 1E CD  3135

0200  2A 5E BF D1 6D 03 37 BA  : 79
0208  B6 83 40 0D 06 FF F7 6E  : F0
```

▶13日の金曜日って不吉ですか？　私の生まれた日がそうだったんですけど（21年前の3月)。ちなみに私の家にはチェーンソーがあったりします。　　倉田　泰幸(21)茨城県

```
0210  19 C4 88 91 37 C7 B1 57  : FC
0218  7B 13 56 81 02 B1 B7 71  : 40
0220  21 25 3B 77 D7 DC AB 74  : CA
0228  AB B0 93 00 C7 27 95 6A  : DB
0230  C0 D3 01 29 6E AE C8 7D  : 1E
0238  96 F0 7E B9 75 0F E0 A7  : C8
0240  4B 82 07 59 B7 FE 35 8C  : A3
0248  E9 E2 AF 11 F0 33 B5 54  : B7
0250  49 96 AE 8C E1 0B 05 C5  : CF
0258  30 88 70 22 FC A7 FF 43  : 2F
0260  F8 44 3E DE AC 90 F1 EA  : 6F
0268  25 8F 10 FD 9B D9 52 68  : EF
0270  74 85 A9 7A 17 70 D3 BC  : 32
0278  E5 A2 1E 4D DC 98 FC 74  : D6
-----------------------------------
SUM:  B9 CC 13 03 EB 8E 7E 5C  B420

0280  E3 02 78 84 60 89 40 10  : 1A
0288  E1 48 BC B7 C4 26 15 BC  : 57
0290  EB 8A 8D C4 F6 F4 92 01  : 43
0298  90 C5 FF 15 8C 71 D3 6B  : A4
02A0  D5 60 53 0B 6C F0 99 79  : 01
02A8  64 3E 8B B0 F1 5F 0A 54  : 8B
02B0  6F BA F1 A7 A2 61 E2 95  : 3B
02B8  A2 62 D4 55 12 41 09 FF  : 88
02C0  1F 7E 87 80 D8 32 E9 F6  : 8D
02C8  95 0D 25 B2 C5 F8 57 EF  : 7C
02D0  C5 D6 DB DE 56 FD EF B9  : 4F
02D8  11 B3 49 37 2D 26 F1 C7  : 4F
02E0  10 AD 9D 24 C7 D6 79 2E  : C2
02E8  C7 E4 27 03 A2 64 BE 82  : 1B
02F0  7C 2E E5 D8 24 73 61 1F  : 7E
02F8  0D 86 67 2F CD 2E B5 21  : FA
-----------------------------------
SUM:  73 AC 43 40 31 2D B5 EE  ED49

0300  9D FB 2F 2F F1 FE 8B 62  : D2
0308  E8 7D D2 46 9D F6 DF C1  : B0
0310  63 D3 52 EC E8 6F 55 0F  : 2F
0318  14 3D 0A 28 8A 1A 60 E1  : 68
0320  97 6F DA EB 23 DF E1 7E  : 2C
0328  BE 40 1C C7 EF CE FC 32  : CC
0330  6A 9B 7E A2 96 0D C9 2F  : C0
0338  FE AE A2 31 F2 95 0A A7  : B7
0340  B3 87 8A FA CF 0B 5B BB  : AE
0348  01 F5 AE B2 B0 70 24 6F  : 09
0350  B6 9C B6 40 F6 9C B8 86  : 18
0358  E8 EA B5 18 FC 02 E1 10  : 8E
0360  5A 6F DE BE A9 AB 5D 13  : 29
0368  DE 4E 4A E9 C5 A1 B5 D9  : 53
0370  A8 82 1F B4 98 BF E1 59  : 8E
0378  B8 93 52 C0 8E AE CD D7  : 3D
-----------------------------------
SUM:  A3 54 AF 2D 9F 9E A7 75  C115

0380  9D BC 78 37 49 B7 E0 DF  : C7
0388  8F 48 D9 AB A6 4D 77 81  : 46
0390  4F E3 FB 8B 1E 5E AD 75  : 56
0398  E5 F9 64 90 3D B3 07 EC  : B5
03A0  9F 6E ED 70 A4 C5 6A 0E  : 4B
03A8  B1 63 07 98 4E D6 48 2C  : 4B
03B0  8A 95 EF 81 1E E2 84 94  : A7
03B8  15 9E 8C 84 47 F6 89 D2  : 5B
03C0  C9 D1 11 E7 D5 92 1F A9  : C1
03C8  CB 14 03 F2 A9 75 28 78  : 92
03D0  9D AD 5B 96 D0 C9 6F F9  : 3C
03D8  2F 5E EE C1 F5 27 E6 94  : D2
03E0  C2 61 D1 5E 25 72 53 F0  : 2C
03E8  59 73 AD F8 36 88 5D DB  : 67
03F0  DE 85 DB 27 EE 98 0C 38  : 2F
03F8  D6 C0 9A F2 BB 89 A1 F8  : FF
-----------------------------------
SUM:  7E ED 6F A9 E8 9A C3 0A  7829

0400  54 3D EB C0 D0 F8 86 79  : 03
0408  94 36 E2 BC 7C 92 97 28  : 35
0410  4D 9F 03 98 A0 03 52 DF  : 5B
0418  83 42 65 2D BE 68 9F B6  : D2
0420  B2 72 42 53 8A F7 BE 3C  : 34
0428  C6 81 63 E4 D4 BE 54 41  : B5
0430  CC 96 6D 85 C9 05 1B C4  : 01
0438  8A 5F B5 1D 80 7E E5 28  : C6
0440  3E 2F 48 3F 8B 9A 0F DC  : 04
0448  58 E1 F4 94 4C 9E A5 C7  : 17
0450  26 36 0B 98 04 82 C7 33  : 7F
0458  8D 52 38 59 15 F0 18 CF  : 5C
0460  0E F9 F0 4C 6D 6D 2F 40  : 8C
0468  C6 99 99 62 5B F1 3F 1A  : FF
0470  AB C0 28 BC C2 0B F0 12  : 1E
0478  5D 15 0B D7 9C 17 86 97  : 24
-----------------------------------
SUM:  AB 3B 37 1F 67 57 97 47  2AB7

0480  0A C3 2F F2 10 E3 DA 9B  : 56
0488  49 40 1A EF 15 DA D5 EC  : 42
0490  36 27 6B 86 DF 5F B8 28  : 6C
0498  60 C4 4A 44 00 F2 A5 B9  : 02
04A0  C7 3B BF F6 FB 96 BC CE  : D2
04A8  3A 9A 8E 66 FC AE D8 63  : AD
04B0  2F 6E BB 65 FE B1 ED E1  : 3A
04B8  C9 AD EF 6C 88 6F 85 95  : E2
04C0  C8 0B B0 E5 D9 50 2F 87  : 47
04C8  D9 44 A5 DF EC 06 D5 FF  : 67
04D0  06 33 CA 66 5F CA 78 31  : 3B
04D8  54 FE 37 78 31 30 17 BA  : 33
04E0  71 6D F5 6D 2F A0 8C 9D  : 38
04E8  BC F7 91 EE 5B 38 E1 1B  : C1
04F0  36 6E BF 1F 9B F6 08 F3  : 0E
04F8  5A BF CB 96 A5 E5 52 97  : ED
-----------------------------------
SUM:  9A EF 5B 8A A0 75 6C C2  82D7

0500  3E BD 7C 16 0E AB F2 B1  : E9
0508  2F 02 34 D2 BA 53 98 B6  : 92
0510  AB 29 D0 F0 CE E3 62 5A  : 01
0518  62 84 31 F9 0C 54 86 B2  : A8
0520  C7 EE EE E7 1A BF 91 01  : F5
0528  D4 B6 BC 61 C9 19 B2 7A  : B5
0530  F6 E2 BB 50 58 E4 45 24  : 88
0538  FB D2 FF B5 0F 75 7A 3E  : BD
0540  D9 FB 6E EF E6 07 15 96  : C9
0548  16 85 BE 8E 8D 2B 4E 6B  : 58
0550  2B 83 A2 2D F0 97 C2 DD  : A3
0558  66 D0 85 F4 0B E4 32 AC  : 7C
0560  A3 43 05 3B BD 9E 0B 3C  : C8
0568  1C EC 35 9C 5D 86 00 2E  : EA
0570  BA 80 66 41 F0 6F 13 4B  : 9E
0578  D6 E4 C1 06 91 C3 AF BD  : 41
-----------------------------------
SUM:  D5 2A C9 DA F5 69 98 4C  97AE

0580  B8 F4 FB 10 3F CD D5 2A  : C2
0588  CA 3B 87 E0 97 D2 09 FB  : D9
0590  11 D9 C4 D0 C4 6E 61 C2  : D3
0598  3B FD A8 8F D4 69 B8 03  : 67
05A0  E6 3B 48 88 FE A5 14 62  : 0A
05A8  F6 6B C3 5D F5 39 9B 1A  : 64
05B0  06 4D 1F 3E 9E 9D C5 7A  : 2A
05B8  2E 03 30 C4 7D 78 5E 37  : AF
05C0  5E B9 93 96 20 33 E4 E8  : 5F
05C8  A9 67 D6 45 75 C1 07 F3  : 5B
05D0  62 0F 5A 08 3D D9 03 D0  : BC
05D8  81 6D 38 65 DA 3D BF B9  : 1A
05E0  9E 1B B1 6D 2F 80 96 DE  : FA
05E8  0A E9 E0 A4 31 9E F0 94  : CA
05F0  99 0E 07 D8 49 C2 F4 40  : C5
05F8  45 2F B1 1A B7 0A E3 8F  : 72
-----------------------------------
SUM:  4E D8 8C 81 88 5D D3 BC  753B

0600  F6 97 FE 1F 9F A6 EF 7B  : 59
0608  E0 80 FC 32 06 37 8F AC  : 06
0610  CA 92 B9 C8 7F 7C D4 E2  : 8E
0618  2F 69 3F C2 82 B4 A2 BE  : 2F
0620  97 26 0F A0 D6 9B 67 B0  : F4
0628  22 92 DC BD 41 8D 1F AD  : E7
0630  C1 FF D2 40 D9 8A B7 7B  : 67
0638  B5 49 49 16 F9 14 C5 F9  : 28
0640  FD 96 DB 54 65 CA 4F 8E  : CE
0648  4C C7 C7 3E 4F 8D EB FF  : DE
0650  DD B7 07 D6 50 51 D6 72  : 5A
0658  73 9E A5 4A 24 4C C3 05  : 38
0660  D1 92 22 4C 2E 93 CB 39  : 96
0668  B3 E3 E2 7D 78 77 7F 3E  : A1
0670  8D A9 6E A0 31 87 DB DB  : B2
0678  B9 EE FD 44 CB 77 76 7B  : 1B
-----------------------------------
SUM:  61 D0 B5 ED 59 CF 64 69  56E2

0680  3C B3 B9 FE BD 54 3F 71  : 67
0688  9C FF D2 7E 37 00 C4 85  : 6B
0690  63 D5 D4 69 00 5D FB 3A  : 07
0698  96 67 D8 35 16 3C C2 23  : 41
06A0  7F 08 7F 7B 3A 80 9B 4F  : 25
06A8  A0 12 2C C8 13 CD D8 13  : 71
06B0  B3 26 09 EB E7 82 40 71  : E7
06B8  82 54 9D E8 06 6A 3F 46  : 50
06C0  10 C8 1E BD 7B 89 01 DC  : 94
06C8  E6 38 4B 23 34 46 65 9B  : 06
06D0  F2 3C 27 7F E7 88 4E C2  : 53
06D8  46 D1 81 41 FF C7 1F E6  : A4
06E0  D2 C3 94 5C 71 98 F8 84  : 0A
06E8  DD 58 88 23 AF EE 56 D0  : A3
06F0  3B 8B 7A 8C 83 8A 22 32  : 2D
06F8  E2 81 1A C2 87 74 28 AE  : 10
-----------------------------------
SUM:  1F B6 49 9D 03 C8 1D BF  8433

0700  54 53 84 A1 11 0A 84 67  : D2
0708  8E 5F D9 60 0A 07 C4 CE  : C9
0710  99 E7 A9 3A 45 A8 BC C7  : D3
0718  5F 9C BE C4 D9 17 BC 84  : AD
0720  2B 88 B7 DC 10 EC 2A C5  : 31
0728  61 0D 0B C2 C3 F1 70 1B  : 7A
0730  CE AC F6 74 EA 6C 65 8C  : 2B
0738  CF E2 F5 A9 2B AE B5 92  : 6F
0740  00 4B 4B 5F 4C B1 C7 19  : D2
0748  87 B3 C7 80 04 37 6E 47  : 71
0750  AA 0B 86 AD F9 A7 B6 A2  : E0
0758  2A 0E B2 9B 87 CB 50 3D  : 64
0760  CA 07 F8 E2 03 CF 8E 00  : 0B
0768  FB 4E 97 23 DE C1 A7 68  : B1
0770  15 B8 99 71 79 F2 F6 09  : 41
0778  17 35 24 86 9B F3 B3 28  : 5F
-----------------------------------
SUM:  4F B1 07 DD E6 96 8D 56  B6AD

0780  91 C9 50 3A 52 81 E7 CA  : 68
0788  21 D9 36 9F 71 32 13 DB  : 60
0790  1E FF 47 D4 7E F9 FA 4F  : F8
0798  12 59 41 E5 7A 09 3E 56  : A8
07A0  FD 95 09 B9 80 ED 9A 5E  : B9
07A8  9C 2D EF 0F 3F 73 05 C2  : 40
07B0  3E EB 7A C6 F4 BB 87 59  : F8
07B8  8A 9F 3B 60 C1 CE 3E 4D  : DE
07C0  EF 77 7D 15 4E 1D FC F6  : 55
07C8  0A B8 E8 0E C7 F9 44 4B  : 07
07D0  60 2C 1C CA 1D EB 33 EE  : 9B
07D8  A9 BE 3C 56 16 B1 52 E4  : F6
07E0  BA D6 89 E8 B9 1E 2D 71  : 76
07E8  61 5F D6 D0 54 F8 E9 29  : C4
07F0  A3 DD 69 A5 F6 33 EB F8  : 9A
07F8  B9 AB 8C F8 74 3E E5 4D  : CC
-----------------------------------
SUM:  BC 1C CC 18 EE D7 41 02  156B

0800  6E 40 FB 20 BA 02 FC 82  : 03
0808  18 2B 95 60 AF 29 00 2B  : 3B
0810  9E 98 55 B6 EF DE 37 A0  : E5
0818  4F 57 17 09 11 8A 53 0F  : C3
0820  5D CA A2 30 06 8E A8 69  : 9E
0828  38 23 D3 94 AF C3 B9 E7  : D4
0830  AE 3B 92 7D 5E 05 29 4E  : D2
0838  33 2A 27 73 8A 2D 5B 6A  : 73
0840  EF 42 6D B7 2D 3A C4 77  : F7
0848  20 0A E7 48 16 E0 FF 19  : 67
0850  44 62 20 5F EF 25 4C 90  : 15
0858  BE 68 17 CA F2 05 FE 43  : 3F
0860  FE 23 F9 42 98 2C D3 E0  : D3
0868  B8 2C DE 6D B8 16 AF 01  : AD
0870  B4 E2 F1 02 74 2F 86 C3  : 75
0878  FC 97 2B CF 7F 34 A5 F2  : D7
-----------------------------------
SUM:  60 8A A8 9B 6D FF 25 5D  67C1

0880  B2 02 D7 E5 E5 DB FF FE  : 2D
0888  8E DA 93 E4 BF D2 89 E5  : DE
0890  AB A3 6B 77 2E 5D 12 A8  : 75
0898  30 3A A0 4B 1A 78 E7 40  : 0E
08A0  A8 F4 35 DC 24 8C AE BA  : C5
08A8  69 5F 2C 59 DE A0 01 A0  : 6C
08B0  9B 30 76 B6 36 F7 CC 33  : 23
08B8  BA 16 7D 1B E2 88 32 A2  : A6
08C0  EB D0 34 F5 7F 5D 2D C8  : B5
08C8  2E 09 EF 38 27 78 4E 63  : AE
08D0  CD 3B 67 7F 72 8A EE 3D  : 15
08D8  A3 E5 8D 39 74 9B 04 DA  : 3B
08E0  BB 66 74 9F 6D C2 C0 53  : 76
08E8  8C 59 BB 92 96 B9 1A 0B  : A6
08F0  43 2A ED 65 F7 A7 37 4E  : E2
08F8  C1 D8 5F 71 1F 8A F6 8F  : 97
-----------------------------------
SUM:  55 0C 5B 7D AB D3 A2 77  03B6

0900  BD ED 24 CF 7D 69 DB 28  : 86
0908  ED F8 3F 03 6A FE 07 68  : FE
0910  02 A6 9A 73 86 D5 B0 AE  : 6E
0918  3B 9F F3 A9 32 2A 33 DD  : E2
0920  97 8B BB D7 39 45 ED BD  : DC
0928  E6 D9 A8 17 79 73 E3 7E  : CB
0930  30 73 01 65 7C 45 32 98  : 94
0938  00 98 0B DF CB 58 02 DA  : 81
0940  D0 05 C5 E9 9D 7D BA 4D  : A4
0948  88 5F 22 E4 5B D9 89 09  : B3
0950  F2 F1 5F 16 1A 0C 1B B9  : 52
0958  05 F5 67 87 83 DD 5C 82  : 26
0960  D5 2D CD DC 01 A1 E2 41  : 70
0968  99 21 0F A2 53 C6 83 5E  : 65
0970  ED 76 FB 68 7F B8 FF 9C  : 98
0978  1E 79 CA D7 35 91 74 D1  : 43
-----------------------------------
SUM:  5C 20 AD 47 35 AA 5B 65  960C

0980  40 D7 47 5C 35 79 E7 2B  : 7A
0988  9B CA B9 86 71 94 7E 0C  : 33
0990  6B C5 67 B4 29 0A DD 9E  : F9
0998  81 A9 39 C9 C0 6A 72 0E  : D6
09A0  5F 65 E5 C6 8C AC 94 CE  : 09
09A8  08 B3 4F 14 97 31 E1 77  : 3E
09B0  8D 73 59 4F 45 B4 5A CB  : C6
09B8  18 0F 86 CA CC F4 69 F7  : 97
09C0  E7 19 F9 D8 3E 5E 5D 6E  : 38
09C8  3F 94 98 63 D3 6E 6F 03  : 81
09D0  5C CE BF 07 DB 2C 74 8A  : F5
09D8  7B 10 FF E1 6F A3 FE 2D  : A8
09E0  6D 40 F6 DA 71 3E 83 90  : 3F
09E8  C1 E9 3A 71 90 3E 1F F8  : 3A
09F0  A9 E7 74 F5 8F B2 F9 70  : A3
09F8  28 98 62 45 FA 40 F8 EB  : 84
-----------------------------------
SUM:  CF DC 08 FA A8 0F BD F5  27C9

0A00  5A 59 8F 5A 1F 63 40 33  : 91
```

▶この前「Oh!X」とドアに書いてある車が走っているのを見た。あれはいったいなんだったんだろう。
松本　貴之(16)愛知県

```
0A08  60 CF F2 20 C9 54 73 2D  : FE
0A10  D9 8D 3C F6 26 EB BB 3B  : 9F
0A18  70 1C B4 C7 9A 2D AB 15  : 8E
0A20  BF 6B 28 AB B8 F1 5C A3  : A5
0A28  66 23 FC 07 49 1B B6 D2  : 78
0A30  5E C0 1D 61 0A AE 9A A6  : 94
0A38  EA A7 C5 43 29 8F 23 6F  : E3
0A40  33 CE D8 04 F9 FC F7 74  : 3D
0A48  5B 03 A5 6E 5F 1D F7 BA  : 9E
0A50  5D CE A1 DC F5 F1 56 BC  : A0
0A58  83 18 9B 3A 8E 1D 7B E7  : 7D
0A60  45 5B 5E 55 66 04 E1 61  : FF
0A68  7E 5D 9B 97 D3 79 11 6E  : D8
0A70  B1 BE DC 29 9A D2 E9 91  : 5A
0A78  16 F2 CF 68 2F AE 38 D9  : 2D
-----------------------------------
SUM:  68 E5 D4 92 B9 3C BA 44  0CB5

0A80  7D 10 47 82 EA 2A DF 13  : 5C
0A88  D9 40 72 43 77 65 A9 DA  : 2D
0A90  8B B0 11 A0 0F E4 36 57  : 6C
0A98  8F 7C 96 48 D9 13 DA D7  : 86
0AA0  62 8E D7 AB 6D 8B F9 E4  : 47
0AA8  67 F5 3C B4 80 FA 33 F4  : ED
0AB0  D1 79 38 35 0E 35 0B 39  : 3E
0AB8  E4 D5 95 EC AC 01 6A B6  : 07
0AC0  2C 93 20 C5 FD 0F 62 7D  : 8F
0AC8  6E 69 8E 4D 7B DA 12 3B  : 54
0AD0  21 66 CD B0 15 DC 9C 61  : F2
0AD8  1A 58 7B 32 B4 60 23 8F  : E5
0AE0  8A BE C6 D0 4D A2 1C 2F  : 18
0AE8  13 F6 07 4B BE 48 A8 C1  : CA
0AF0  36 DA 9E 68 0E 89 CE 2F  : AA
0AF8  0B 9F F1 3E A1 55 85 30  : 84
-----------------------------------
SUM:  A1 34 92 E2 EB 2E 83 D9  2F64

0B00  83 35 67 19 E5 FD C8 8E  : 70
0B08  EA F9 95 B1 5B F3 C6 8F  : CC
0B10  43 28 B3 30 A4 41 C8 75  : 70
0B18  F7 C0 04 E7 48 CC FD 99  : 4C
0B20  83 BA DD F6 D2 A6 78 D9  : D9
0B28  A4 E6 93 E8 37 2B BB AD  : CF
0B30  0B A6 97 0B B0 EF DD C1  : 90
0B38  7C A7 A9 A7 2A 5D 4B 2B  : 70
0B40  B9 33 88 08 FC F6 76 67  : 4B
0B48  E1 1D F9 39 A5 79 74 B8  : 7A
0B50  04 16 B3 84 7A AA 91 79  : 7F
0B58  1D 25 DA E8 52 39 85 21  : 35
0B60  B9 DB F2 50 0F 32 97 A3  : 51
0B68  57 EF D4 9D C7 11 23 B0  : 62
0B70  D3 0C 25 27 7A 60 88 FE  : 8B
0B78  37 52 19 49 F2 4C F3 0A  : 26
-----------------------------------
SUM:  2A B6 75 7B BE 5B E3 B1  D52A

0B80  4C 93 2C 55 27 76 92 8B  : 1A
0B88  EB 5C 35 26 85 CB 2A 93  : AF
0B90  BD 29 46 6D 63 29 1B D9  : 19
0B98  F5 AD 74 35 54 A4 F9 D0  : 0C
0BA0  A4 D8 04 EA 0D E1 63 3A  : F5
0BA8  75 E7 92 7B 8C 0A FA CF  : C8
0BB0  75 3A AA 3F 25 9A 06 CB  : 28
0BB8  07 41 B4 4A 76 99 DF AD  : E1
0BC0  16 54 73 30 CF 7A 5E B7  : 6B
0BC8  CC F4 90 82 F7 56 0A 17  : 40
0BD0  F2 5B 55 EC 7F E3 AF 87  : 26
0BD8  B2 7E 00 95 72 AD D9 04  : C1
0BE0  09 EC 20 59 26 55 5F E6  : 2E
0BE8  77 CE 71 BC 72 43 AF EE  : C4
0BF0  F4 0E 71 F7 EA 36 0B 93  : 28
0BF8  39 55 A5 6D 3D 1C 83 6A  : E6
-----------------------------------
SUM:  B1 3D 0E B7 0D 76 9E 72  BB8F

0C00  3F E6 AA B7 70 34 C5 25  : 14
0C08  E5 A4 99 24 C1 47 9D D0  : BB
0C10  2C 24 F0 FA E9 3B 81 02  : E1
0C18  1E 8B 1C 3E D0 46 FA 9F  : B2
0C20  97 E1 01 C3 95 EF 52 6D  : 7F
0C28  B7 D9 91 25 58 B2 DC 0A  : 36
0C30  50 76 05 0C 09 D1 24 FC  : D1
0C38  96 C7 E9 EF A7 B7 03 AD  : 43
0C40  4E D6 00 E7 F4 D7 F6 FE  : CA
0C48  D0 E0 A5 BF 25 70 9D F4  : 3A
0C50  C5 7B 4D 62 03 60 3F C8  : 59
0C58  F7 C6 AE 57 C8 63 74 38  : 99
0C60  15 56 41 22 29 04 38 B1  : E4
0C68  78 22 19 7F 72 44 35 D3  : F0
0C70  26 FA 9F 3E BA 6B 66 EE  : 76
0C78  6E 68 1F 4C 56 98 1D 2F  : 7B
-----------------------------------
SUM:  9D 01 87 80 16 7A 68 49  1DA1

0C80  7A 56 03 E6 F5 F6 10 3C  : F0
0C88  77 DF C4 07 EE 2D 32 F5  : 63
0C90  CE FD CF A6 87 98 0F E2  : 50
0C98  8E 60 3B 7E 37 60 3E 87  : 03
0CA0  E5 B7 CB BB DA A9 02 9E  : 45
0CA8  03 8D E0 0F F6 54 AA 0E  : 81
0CB0  40 9A A3 2A 5C 0F 21 52  : 85
0CB8  E0 79 0A 97 03 C8 54 B8  : D1
0CC0  1E 72 A6 00 E4 2A 6E E1  : 93
0CC8  15 24 03 90 A9 28 1C E5  : 9E
0CD0  4A 00 F3 95 3C EE 51 52  : 9F
0CD8  80 38 D1 E8 8B 93 7A E2  : EB
0CE0  51 96 B0 03 A6 6B 95 D3  : 13
0CE8  EC 8B A7 62 98 59 39 CC  : 76
0CF0  C7 17 72 CF D2 62 A0 E3  : D6
0CF8  37 6C FF 25 57 3E F2 41  : 8F
-----------------------------------
SUM:  8D 5B 5E 02 8B 26 65 0D  4EC5

0D00  71 A3 8C BD 01 9A 64 27  : 83
0D08  A9 95 09 6A 43 AE 99 D2  : 0D
0D10  6C E6 DF C8 C0 57 A3 73  : 26
0D18  33 9B 0A 16 ED 9A C1 16  : 4C
0D20  84 BA D6 9B 00 18 00 5D  : 24
0D28  9B 76 B0 3E AE D0 3E AE  : 69
0D30  0B BE 85 DB B0 62 62 F8  : 95
0D38  80 64 01 E1 A0 F1 AE 80  : 85
0D40  4E 2A C1 2C 40 27 95 60  : C1
0D48  90 EF 87 5B AC 01 F5 93  : 96
0D50  0D 30 2D 80 38 89 B3 D0  : 2E
0D58  81 DD 93 15 78 0D C7 C2  : 14
0D60  F1 A4 19 59 3E B5 C1 20  : DB
0D68  0B C1 BB 73 6D 7C 3A AF  : CC
0D70  44 5B 63 74 A8 FD 9B 39  : EF
0D78  90 40 6E 99 78 6F 04 B0  : 72
-----------------------------------
SUM:  9F 31 37 8F 56 CF 4D 42  35BC

0D80  59 42 32 52 C8 47 78 1C  : C2
0D88  80 46 F1 1F 0E D0 8E 4B  : 8D
0D90  F9 08 F3 EE 48 DF A6 B0  : 5F
0D98  02 F7 69 68 0B E0 67 E7  : 03
0DA0  80 8F F5 57 7D 01 E6 BC  : 7B
0DA8  0A EE A0 53 B6 02 DF F7  : 79
0DB0  02 ED B9 8B 87 FD B1 4D  : B5
0DB8  9F FA E8 75 F0 F4 C8 15  : B7
0DC0  AF 0F 1A 2D B2 31 E5 2E  : FB
0DC8  FA 2A 50 A5 93 EC 86 E1  : FF
0DD0  5B 30 C1 73 E3 54 27 3C  : 59
0DD8  91 B7 E5 CF 55 5D 4E 2B  : 27
0DE0  ED 0C AD 3B 76 C0 5F 3D  : B3
0DE8  7E 36 6B 94 2F 6A F1 EB  : 28
0DF0  67 C2 50 6A B1 B5 FB 1A  : 5E
0DF8  47 7F 6A 31 B8 A9 53 61  : 76
-----------------------------------
SUM:  AD 8E 97 EF 5E 20 CF 2C  1DC0

0E00  CA 47 43 CF 71 1F 3E 6C  : 5D
0E08  5B D4 2B AF 25 CD 3C FB  : 32
0E10  C5 A6 3E 23 79 17 4C BF  : 67
0E18  81 93 63 6D B3 E9 9C 19  : 35
0E20  5A F7 57 C3 0C 63 AD B7  : 3E
0E28  D6 BF 25 5E 74 EF 76 0C  : FD
0E30  E0 A5 16 6F FE E4 87 FE  : 71
0E38  1B A8 3D A3 B8 D6 DD CE  : DC
0E40  C1 A2 7A 8B 81 4B AB 4B  : 2A
0E48  16 22 EE BA 9E EB 5A FD  : C0
0E50  89 C2 B7 DE A9 BB B9 B2  : AF
0E58  2D 79 F7 4F 3A 78 2F 08  : D5
0E60  1D 16 3D FF F5 63 FA 90  : 51
0E68  46 D3 03 D9 EA 2C 87 A8  : 3A
0E70  F8 08 47 FC A7 13 5B E0  : 38
0E78  D9 AC F3 55 B4 D0 70 FD  : BE
-----------------------------------
SUM:  57 F3 6E DC 34 D3 22 E5  DCD1

0E80  DA 09 79 73 5F F3 A1 47  : 09
0E88  F1 7F 09 C4 C4 96 DD 72  : E6
0E90  75 1F 9E 69 61 F4 A3 D5  : 68
0E98  65 66 8D C7 86 61 2F 56  : 8B
0EA0  9F E3 59 78 A8 15 F7 EC  : F3
0EA8  C5 71 A9 D6 8C 89 92 BA  : 16
0EB0  0B C7 CF 7B 43 7D CA B8  : 5E
0EB8  07 9D 84 FC C5 C0 14 66  : 23
0EC0  15 DF A3 CF 02 1D 39 F6  : B4
0EC8  80 71 5F B4 9E 41 DD 6C  : 2C
0ED0  40 26 5F D6 78 0A F9 18  : 2E
0ED8  59 CE A7 78 2B BA 36 6E  : CF
0EE0  64 C7 8E 76 68 D2 DF 3B  : 83
0EE8  7E 53 B4 55 CD 50 ED 67  : 4B
0EF0  E4 34 D5 8E A2 A2 F1 16  : C6
0EF8  FE AF AA 2F FE AB 1B 6B  : B5
-----------------------------------
SUM:  0D 06 CB 85 5E 4A D4 B3  C2FB

0F00  44 1F 9C FE BC C5 8D 88  : 93
0F08  9A 84 DA B0 15 9F 95 AF  : A0
0F10  39 74 B1 A0 17 92 B6 03  : 60
0F18  7C 64 F5 0A 79 0B B5 E0  : F8
0F20  76 A1 39 0D 7C E2 A6 DD  : 3E
0F28  A5 6E B3 7C 0B 8D 5D 0B  : 42
0F30  76 5A 12 A8 97 74 F8 06  : 93
0F38  A9 80 F8 C0 18 84 07 A8  : 2C
0F40  01 99 73 B5 7D C4 6B F2  : 60
0F48  B3 DE 8E E1 3F 60 0A 85  : 2E
0F50  EE 7B 3D 03 D7 4C 66 7B  : AD
0F58  F1 60 34 F5 44 7F D0 20  : 2D
0F60  13 4F 3B 3D 06 40 23 67  : AA
0F68  EA 63 63 21 9B C7 C2 6E  : 63
0F70  25 32 88 68 F0 CE E6 C3  : AE
0F78  3B 72 96 56 3F 3D 94 AE  : 57
-----------------------------------
SUM:  BD 0C 40 F3 3E 69 99 08  645D

0F80  B2 95 8C BD F4 87 7E 9A  : 23
0F88  FE 0E 30 8D E8 E5 FA F9  : 89
0F90  B9 83 D8 D7 2C 82 D7 F5  : 65
0F98  E9 9F 31 DF 58 FB B4 8F  : 2E
0FA0  9E 65 22 97 74 CE 69 02  : 69
0FA8  EC 51 48 15 A7 50 90 2A  : 4B
0FB0  E8 EA A0 2E 75 C0 BF 13  : A7
0FB8  26 29 05 54 93 A2 83 BE  : 1E
0FC0  8D 57 F5 C1 CF 79 DE 47  : 07
0FC8  DD A7 89 6E 9C 76 B2 28  : 67
0FD0  14 59 D5 1F 8A B2 58 55  : 4A
0FD8  19 43 C5 9A 21 0F 62 47  : 94
0FE0  47 7A 09 AB 17 4A 9D 16  : 89
0FE8  4A 4E B5 29 12 B4 61 0A  : A7
0FF0  7F B3 4D 9E 50 D8 BB 91  : 91
0FF8  48 B5 73 78 14 7C 17 61  : F0
-----------------------------------
SUM:  D9 58 6A 00 26 6B 58 31  7E3B

1000  A4 40 BA 77 AC A9 CF C1  : FA
1008  6A 7C 46 7D DC FF 54 DE  : B6
1010  E4 A8 E6 AD 20 05 98 EA  : C6
1018  4A 89 21 4A EE 18 92 2F  : 05
1020  CD 5B 87 82 01 EF CF FA  : EA
1028  17 C4 45 B7 F8 0E 85 64  : C6
1030  96 5E 72 19 CB DB 46 5E  : C9
1038  EF DF B7 08 16 50 49 51  : 8D
1040  09 79 1E 3B 8E B7 AA 64  : 2E
1048  93 90 EF EE 1F E4 89 E9  : 75
1050  06 38 44 B7 BF BC F8 87  : 33
1058  0A 78 75 B8 3B 3E BE 4A  : 30
1060  60 68 8E 54 29 D4 90 C4  : FB
1068  D3 EF 01 C2 67 00 B5 39  : DA
1070  78 2D DB 5C 2D 6E 61 20  : F8
1078  F1 73 16 2B 7E 66 CE 1F  : 76
-----------------------------------
SUM:  ED F9 42 7A 52 2A 8D 1F  B9FA

1080  B8 A5 F1 99 22 C6 B2 85  : 06
1088  22 EE 8E 2B 71 C0 E0 53  : 2D
1090  E6 92 1B 28 28 43 C2 97  : 7F
1098  EF 46 47 D5 68 B2 3B AF  : 55
10A0  A6 34 9F 1B F4 81 3E 6C  : B3
10A8  F8 59 30 34 E3 42 8F C8  : 31
10B0  C4 77 4A 4C 7A 29 25 FB  : 94
10B8  75 B3 14 B4 C9 9D 5D 17  : CA
10C0  20 BC 26 9C 6C 91 75 E1  : F1
10C8  49 00 FA 7F D1 6A 19 C3  : D9
10D0  70 51 08 31 9C 6D A2 0D  : B2
10D8  B6 19 45 0A 1B 64 3D 43  : 1D
10E0  4D D3 06 E2 0A BD 20 B4  : A3
10E8  77 1B 23 34 6C B1 B0 E8  : 9E
10F0  17 15 0B D8 4E 05 53 E8  : 9D
10F8  14 C6 E2 FA F0 75 0A 00  : 25
-----------------------------------
SUM:  04 11 91 4E E5 B8 78 DC  B0B7

1100  16 F3 2D B4 B8 31 84 5B  : B2
1108  0B 09 4A 09 54 F8 13 E2  : A8
1110  DE 0C D9 F9 9B 1F 6D 67  : 4A
1118  62 7D 9E 3B DD 52 4D F0  : 24
1120  00 63 FC 8A 0E 80 07 C4  : 42
1128  DB E0 9E BC 4C BE 01 77  : 97
1130  62 D5 C1 F7 40 07 F8 C5  : F3
1138  19 F2 03 95 4D F1 49 0B  : 35
1140  A9 F3 44 88 05 C8 A2 78  : 4F
1148  01 A6 40 4F 6D 00 9D F0  : 30
1150  17 5D 78 17 33 CD 4E 01  : 52
1158  F9 77 87 6F 52 5F 46 31  : 8E
1160  94 38 03 C8 67 8C 80 76  : 80
1168  81 7F 48 0F CE 45 BB 92  : B7
1170  07 DE 4F DE 19 61 38 0F  : D3
1178  8C E4 56 7C 3B 90 0E EF  : 0A
-----------------------------------
SUM:  19 75 BF 51 EB 86 EE 3F  E737

1180  7C 0F 97 BE 07 38 D4 E0  : D3
1188  15 54 A5 90 DC 59 31 E1  : E5
1190  C8 FF 5B 90 FD E9 01 F4  : 8D
1198  48 5D 05 7A DB 66 6D 52  : 24
11A0  91 2E A5 9D 86 E3 58 B8  : 7A
11A8  00 FD EC 7B 91 2A 0A FD  : 26
11B0  A0 F8 75 86 BA C7 90 3D  : E1
11B8  CC 50 9D C8 5B 77 80 50  : 23
11C0  2D 83 30 0E F9 24 FD 10  : 18
11C8  CF 92 A4 23 E3 15 4A 90  : FA
11D0  1F 10 FA 53 FE E8 0F 00  : 71
11D8  B0 1F C7 A8 1F FC 01 83  : DD
11E0  ED 9F EF D1 29 07 DC 23  : 7B
11E8  6D 24 0F B8 04 00 00 00  : 5C
11F0  00 00 00 00 00 00 00 00  : 00
11F8  00 00 00 00 00 00 00 00  : 00
-----------------------------------
SUM:  C3 39 D2 73 0D 4F 18 8F  34EA
```

▶免許の書き換えに行ってきました(普通免許)。午前の部,11時過ぎてから無理やり書類をもらって写真などを撮り新しくなった免許証を手にしたのは11時45分。やっぱり更新センターはすごい。

菅谷　英明(24)兵庫県

表2

コマンドの送り方

A0レジスタにコマンド列の格納アドレスを入れて，$FD13ファンクションをコールする。

```
例：lea.l   command,a0
    dc.w    $FD13
```

$0000 LINE

連続した直線を描く

＜データ＞	＜バイト数＞	＜内容＞
$0000	2	コマンド
n	2	座標総数
X_1	2	座標1
Y_1	2	
⋮	⋮	⋮
X_n	2	座標n
Y_n	2	

＊描画色を$0010 (COLOR) コマンドで設定する

以下，2バイトの座標データは−32768〜32767 ($8000〜$7FFF)

$0001 SPLINE

3点を結ぶスプライン曲線を描く

＜データ＞	＜バイト数＞	＜内容＞
$0001	2	コマンド
X_1	2	座標1
Y_1	2	
X_2	2	座標2
Y_2	2	
X_3	2	座標3
Y_3	2	

＊描画色を$0010 (COLOR) コマンドで設定する

$0002 BOX

2点を対角線とする長方形を描く

＜データ＞	＜バイト数＞	＜内容＞
$0002	2	コマンド
X_1	2	座標1
Y_1	2	
X_2	2	座標2
Y_2	2	

＊描画色を$0010 (COLOR) コマンドで設定する

$0003 TRIANGLE

三角形を塗りつぶす

＜データ＞	＜バイト数＞	＜内容＞
$0003	2	コマンド
X_1	2	座標1
Y_1	2	
X_2	2	座標2
Y_2	2	
X_3	2	座標3
Y_3	2	

＊描画色を$0010 (COLOR) コマンドで設定する

$0004 BOX FULL

2点を対角線とする長方形を塗りつぶす

＜データ＞	＜バイト数＞	＜内容＞
$0004	2	コマンド
X_1	2	座標1
Y_1	2	
X_2	2	座標2
Y_2	2	

＊描画色を$0010 (COLOR) コマンドで設定する

$0005 CIRCLE FULL

円を塗りつぶす

＜データ＞	＜バイト数＞	＜内容＞
$0005	2	コマンド
X_1	2	中心座標
Y_1	2	
R	2	半径

＊描画色を$0010 (COLOR) コマンドで設定する

$0006 SET WINDOW

ウィンドウを設定する

＜データ＞	＜バイト数＞	＜内容＞
$0006	2	コマンド
X_1	2	左上座標
Y_1	2	
X_2	2	右下座標
Y_2	2	

$0007 SET MODE

グラフィックモードを設定する

＜データ＞	＜バイト数＞	＜内容＞
$0007	2	コマンド
MODE	2	ラインモード

ラインモード 0:PRESET 1:XOR 2:OR 3:NOP

備考：デフォルトはORモード

$0008 ＊RESERVE＊

$0009 CLS

ウィンドウ内をクリアする

＜データ＞	＜バイト数＞	＜内容＞
$0009	2	コマンド

$000A ＊RESERVE＊

$000B SET 3D PARAMETER

3D−2D変換用のパラメータを設定する

＜データ＞	＜バイト数＞	＜内容＞
$000B	2	コマンド
P num	2	パラメータナンバー
DATA	2	設定データ

パラメータナンバー

0：CX 1：CY 2：CZ	物体の位置（オフセット）
3：DX 4：DY 5：DZ	回転の中心座標
6：HEAD 7：PITCH 8：BANK	回転角（度）

＊物体の位置は，MAGIC内部では次のSET 3D DATAで指定した座標に加算して処理される。

$000C SET 3D DATA

物体の3Dデータを設定する

＜データ＞	＜バイト数＞	＜内容＞		
$000C	2	コマンド		
PCT	2	頂点の数（n）		
X_1	2	頂点1	頂点リスト	
Y_1	2			
Z_1	2			
⋮	⋮	⋮		
X_n		頂点n		
Y_n				
Z_n				
LCT	2	線分の数（m）		
COLOR	（2）	カラーコード拡張用		
LS_1	2	線分1	線分リスト	
LE_1	2			
⋮	⋮	⋮		
LS_m	2	線分m		
LE_m	2			

＊ひとつの線分は2つの頂点ナンバーで指定する

＊COLORはCRTコマンドで拡張モードを指定した場合に必要なデータ

$000D TRANSLATE 3D→2D

3Dデータを3Dパラメータに従って2Dに変換し，ワークエリアに格納する

＜データ＞	＜バイト数＞	＜内容＞
$000D	2	コマンド

$000E DISPLAY 2D

TRANSLATE 3D→2Dで変換したデータに従ってワイヤフレーム表示する

＜データ＞	＜バイト数＞	＜内容＞
$000E	2	コマンド

＊$000B,$000C,$000Dコマンドの実行後に行う必要があるが，ワークエリアの内容が破壊されていなければ，このかぎりではない

$000F DONE

MAGICを終了し，呼び出したシステムに戻る

＜データ＞	＜バイト数＞	＜内容＞
$000F	2	コマンド

$0010 COLOR

LINE,SPLINE,BOX,TRIANGLE,BOX FULL,CIRCLE FULLコマンドの描画色を設定する

＜データ＞	＜バイト数＞	＜内容＞
$0010	2	コマンド
COLOR	2	カラーコード

$0011 CRT

画面モードを設定する

＜データ＞	＜バイト数＞	＜内容＞
$0011	2	コマンド
CRT_MODE	2	画面モード

CRT-MODE：画面の解像度

0：256×256 256色 2面
1：512×512 256色 2面
2：768×512 16 色 1面
3：256×256 256色 4面
4：512×512 256色 4面

上記の値に$100を足すと，拡張モードになる

＊起動後必ず設定する必要がある

備考：グラフィックページ4面モードでは，ワイヤフレーム表示でページ0,1のみ使用し，ページ2,3は影響を受けない。

$0012 INIT

MAGICのワークエリアの初期化

＜データ＞	＜バイト数＞	＜内容＞
$0012	2	コマンド

＊起動後必ず実行する必要がある

$0014 APAGE

LINE, SPLINE, BOX, TRIANGLE, BOX FULL, CIRCLE FULLコマンドの書き込みページを設定する

＜データ＞	＜バイト数＞	＜内容＞
$0014	2	コマンド
PAGE	2	ページ番号

＊デフォルトはページ0

$0015 DEPTH

ワイヤーフレーム表示の投影面と後方面を設定する

＜データ＞	＜バイト数＞	＜内容＞
$0015	2	コマンド
minz	2	投影面Z値
maxz	2	後方面Z値

＊デフォルトは投影面50，後方面2000

MAGIC.FNCの使い方

3D関数の基本操作

Miki Noritaka 御木 徳高

MAGICを使うのにアセンブラは必ずしも必要ではありません。MAGIC.FNCを使えばX-BASICからでも手軽に扱うことができるのです。ここではMAGIC.FNCの基本的な使い方を実例を交えて紹介してみましょう。

突然ですが，皆さんはMAGICをどのように使われているでしょうか。えっ，SIONで遊ぶときだけ？　そんなこといわずに，せっかくBASICでも使えるようになったんだから活用しましょう。というわけで，MAGIC.FNCの主に3D関係について簡単に説明していきます。

で，7月号掲載のMAGIC.FNCはもう打ち込んだでしょうか？　あ～そうそう，7月号では「アセンブラやリンカを持っていなくてもMAGICが使える」と書いてありましたが，MAGIC.Sをアセンブルするには当然アセンブラ，リンカとインクルードファイルが必要です。ソースを打ち込んだあとで途方に暮れた人もいたんじゃないでしょうか？（いないってそんな奴は）今回はダンプリストで掲載していますのでそちらを入力してください。

まずは下準備

BASICを立ち上げる前に7月号で改良されたMAGICか今月改良されたMAGICを組み込んでおいてください。5月号のままでは一部うまくいかないところがあります。コマンドラインから，

A>MAGIC

でOKです。それからBASIC.CNFに，

FUNC＝MAGIC

を追加し，BASIC.CNFと同じディレクトリにMAGIC.FNCを置いてください。これでMAGIC.FNCが登録されましたので，BASICを立ち上げてください。

3D表示の手順

X-BASICで3Dの物体を表示するときの手順は，

1) magic_flush(　)でデータバッファを初期化する
2) magic_init(　)で3Dワークを初期化する
3) magic_screen(　)で画面モードを設定する
4) magic_color(　)で描画色を設定する
5) magic_putbuf(　)で3Dデータをバッファに置く
6) magic_seek(　)で3Dデータの先頭にポインタを移動する
7) magic_data(　)でバッファ内のデータが3Dデータであることを宣言する
8) magic_para(　)で変換パラメータを設定する
9) magic_pers(　)で3D→2D変換する
10) magic_disp(　)で表示する

という感じですが，順番は必ずしもこの限りではありません。1),2),3)は最初に実行しておけばあとは必要ないでしょう。ここで「バッファ」とか「ポインタ」という言葉について簡単に説明しておきましょう。ここでいうバッファというのはMAGICが直接読むことのできるデータ領域のことです。そしてどこを読んでいるかを示すものをポインタといいます。いま，ここにバッファという名のノートがあるとしましょう。このノートに文字を書くにはページをめくらなければなりません。そして書き終わったときには，一番最後のページが開かれているはずです。それをそのままMAGICに渡してもMAGICにはどこがデータの先頭かわかりません（というより現在開かれているページを先頭だと判断する)。そこでデータの先頭のページを開いてやる必要があるわけです。この操作が「ポインタを移動する」ということなのです。MAGICの内部ではデータはshort int型[*1]（16ビット）です。ところがこのノートには1ページに8ビット(1バイト）しか書けません。よってデータ1個で2ページ，すなわちポインタが2つ進むことになります。

とまあ，ちょっと違うかもしれませんが，こんな感じで理解してください。どうしても理解できない人はお約束だと思ってmagic_putbufの後ろにはmagic_seekがくるものだと覚えてください。変換パラメータはあとで詳しく説明します。その他のコマンドについては7月号90ページのMAGIC.FNCリファレンスを参照してください。

＊1 BASIC上ではint型（32ビット）ですが，MAGICへは16ビットで渡されます。だってX-BASICには16ビット型変数がないんだもん。

表示してみよう

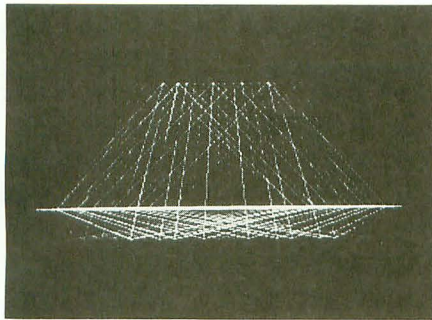
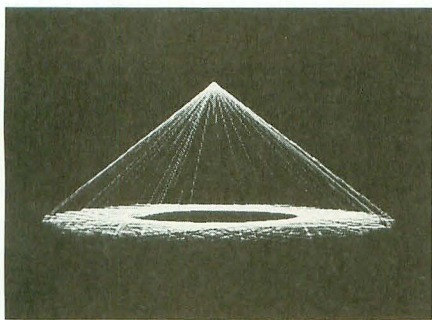
表示の手順がわかりましたので，実際に表示してみましょう。リスト1を打ち込んで実行してください。三角錐が表示されましたね。プログラムでは手順どおりに行っていますので，わかりにくいところはないと思います。3Dデータは配列変数aで与えています。データの形式は表1のようになります。頂点の数や線分の数を間違えると暴走するおそれがあるのでくれぐれも注意してください。

210行でパラメータをクリアしています。パラメータは一度設定すると再設定されるまで残りますので，なにかやる前にいったんクリアしておくのが無難でしょう。

動かしてみる

表示ができたので次はアニメーションに

表1 3Dデータのフォーマット

```
a = {  n                        頂点の数
       X0, Y0, Z0,              頂点0の座標
       X1, Y1, Z1,              頂点1の座標
         :
       Xn-1, Yn-1, Zn-1,        頂点n-1の座標
       m,                       線分の数
       S0, E0,                  線分0の始点と終点の頂点No.
       S1, E1,                  線分1の始点と終点の頂点No.
         :
       Sm-1, Em-1               線分m-1の始点と終点の頂点No.
    }
```

これが基本形の平行移動

回転だ

挑戦しましょう。手順としてはちょっとずつパラメータなどを変えたものを次々と表示すればいいのですが，ここでさっきから無視し続けてきたパラメータについて説明しましょう。パラメータは3D→2D変換するときにMAGICの内部で処理されるもので，物体の移動や回転を行うためのものです。

パラメータナンバー0〜2が物体の原点の位置を指定するもので，リスト1にあったmagic_para(3,400)というのは「物体の原点をZ=400の位置に移動しなさい」という意味です。

同じようにパラメータナンバー6〜8は物体の回転を指示します。そんじゃあ3〜5はなんだ？　というと，これは回転の中心座標（物体の座標系での）を指定するんです。まあ説明するより実際にやってみたほうがわかりやすいでしょう。

●その1・平行移動

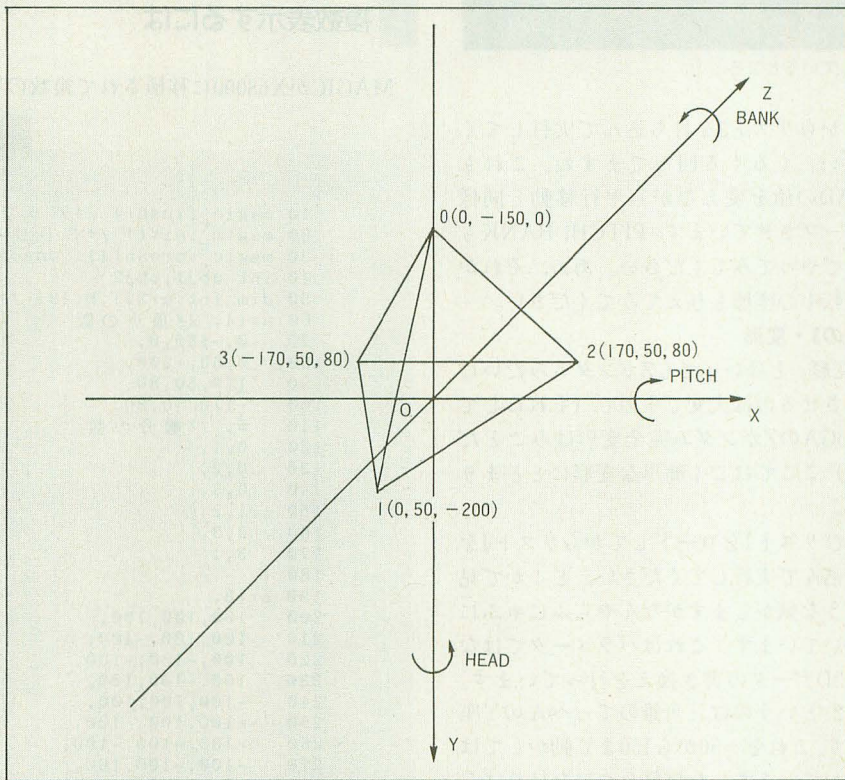
まずは簡単なところから始めましょう。リスト1にリスト2を追加してください。おっと，その前にリスト1はベースになりますのでセーブしておいてください。打ち込んだら実行してみましょう。う〜ん，三角錐が反復横跳びをしている。プログラムではX座標を−100から100まで10おきに動かして，手順7)〜10)をループさせています。7)をループに入れるのを忘れないようにしてください。ここではX方向にしか動かしませんが，各自でY，Z方向にも動かしてみてください。

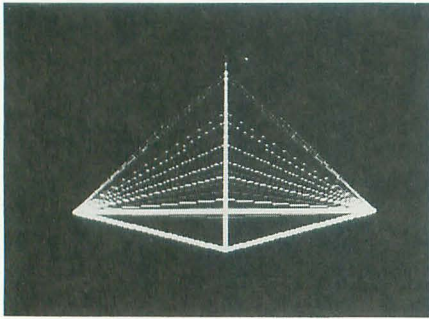
●その2・回転

次は回してみましょう。リスト1をロード

座表系

MAGICの座標系は図1のように右がX，下がY，奥がZとなっています。数学でよく使う座標系とはYとZが逆ですので気をつけてください（でも，右手座標系にはなっています）。それからX，Y，Z軸回りの回転がそれぞれPITCH，HEAD，BANKです。MAGICでは3Dデータや変換パラメータはこの座標系で記述されますので，しっかり覚えてください。背面飛行させたり逆噴射させたりしないように。

図1

リスト1

```
10 magic_flush() /*バッファの初期化
20 magic_init() /*3Dワークの初期化
30 magic_screen(1):console 0,31,0 /*表示画面512×512
40 int obj
50 dim int a(25) /*オブジェクトデータ
60 a={4, /*頂点の数
70   0,-150,0,
80   0,50,-200,
90   170,50,80,
100  -170,50,80,
110  6, /*線分の数
120  0,1,
130  0,2,
140  0,3,
150  1,2,
160  2,3,
170  3,1
180 }
190 magic_color(252) /*描画色
200 for i=0 to 8
210   magic_para(i,0) /*3Dパラメータクリア
220 next
230 obj=magic_putbuf(2,a) /*データをバッファに格納
240 magic_seek(2,obj,0) /*データの先頭にポインタを移動
250 magic_data(2) /*物体定義
260 magic_para(2,400) /*Z座標
270 magic_pers() /*3D→2D変換
280 magic_disp() /*2D表示
290 end
```

変形しているところ

してからリスト3を打ち込んで実行してください。くるくる回ってますね。これもHEADの値を変えながら平行移動と同様にループさせています。PITCH, BANKも各自でやってみてください。ああ，それから回転中心座標も与えてみてください。

●その3・変形

「変形」とはいってもZガンダムみたいに変形させるのは大変ですので（それにしてもDōGAのZガンダム完全変形はみごとだった）ここではごく簡単な変形にとどまります。

再びリスト1をロードしてからリスト4を打ち込んで実行してください。どこかで見たような気がしますがなんやらふにゃふにゃ動いています。これはパラメータではなく，3Dデータの書き換えを行っています。

a(2)というのは三角錐のてっぺんのY座標です。これを−50から150まで動かしては表示しています。ただ気をつけなければならないのは平行移動や回転とはループの開始位置が違うということです。a(2)を書き換えてもバッファを書き換えたことにはならないので，再びバッファにデータを転送する必要があるのです。そのために少し重くなってしまうのはSIONをやっていても気づいたんじゃないでしょうか。

とまあアニメーションの手法としてはこんなもんじゃないでしょうか。もちろん合わせ技も可です。いろいろと試してみてください。

複数表示するには

MAGICがX68000に移植されて複数の物体を一度に表示できるようになりました。ゲームなどではたいてい敵は複数であることを考えると，非常にありがたいことです。方法としては手順5)で物体をまとめてバッファに置いて，6)から9)までを表示したい数だけ繰り返せばいいのです。リスト5を参考にしてください。

いちいちパラメータをクリアする必要はないのですが，前にもいったようにそのほうが無難です。同じものを分身させるだけならmagic_putbufは最初に一度行えばそ

リスト2

```
 35 int x,dx=10
242 repeat
244 x=x+dx
246 if x=100 then dx=-10
248 if x=-100 then dx=10
265 magic_para(0,x) /*X座標
285 until inkey$(0)<>""
```

リスト3

```
 35 int r
242 repeat
244 r=r+3
246 if r=360 then r=0
265 magic_para(6,r) /*HEAD角
285 until inkey$(0)<>""
```

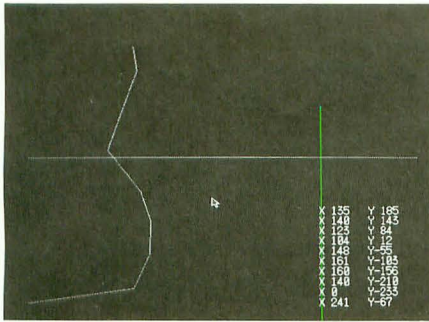
リスト4

```
 35 int h=10
222 repeat
224 a(2)=a(2)+h /*三角錐の高さ
226 if a(2)=50 then h=-10
228 if a(2)=-150 then h=10
285 until inkey$(0)<>""
```

リスト5

```
 10 magic_flush() /*バッファの初期化
 20 magic_init() /*3Dワークの初期化
 30 magic_screen(1):console 0,31,0 /*表示画面512×512
 40 int obj1,obj2
 50 dim int a(25),b(49) /*オブジェクトデータ
 60 a={4, /*頂点の数
 70    0,-150,0,
 80    0,50,-200,
 90    170,50,80,
100    -170,50,80,
110    6, /*線分の数
120    0,1,
130    0,2,
140    0,3,
150    1,2,
160    2,3,
170    3,1
180 }
190 b={8,
200    100,100,100,
210    100,100,-100,
220    100,-100,-100,
230    100,-100,100,
240    -100,100,100,
250    -100,100,-100,
260    -100,-100,-100,
270    -100,-100,100,
280    12,
290    0,1,
300    1,2,
310    2,3,
320    3,0,
330    0,4,
340    1,5,
350    2,6,
360    3,7,
370    4,5,
380    5,6,
390    6,7,
400    7,4
410 }
420 magic_color(252) /*描画色
430 for i=0 to 8
440    magic_para(i,0) /*3Dパラメータクリア
450 next
460 obj1=magic_putbuf(2,a) /*データ1バッファに格納
470 magic_seek(2,obj1,0) /*データ1の先頭にポインタを移動
480 magic_data(2) /*物体1定義
490 magic_para(0,200) /*X座標
500 magic_para(2,400) /*Z座標
510 magic_pers() /*3D→2D変換
520 for i=0 to 8
530    magic_para(i,0) /*3Dパラメータクリア
540 next
550 obj2=magic_putbuf(2,b) /*データ2をバッファに格納
560 magic_seek(2,obj2,0) /*データ2の先頭にポインタを移動
570 magic_data(2) /*物体2定義
580 magic_para(0,-200) /*X座標
590 magic_para(2,400) /*Z座標
600 magic_pers() /*3D→2D変換
610 magic_disp() /*2D表示
620 end
```

これを回すと

こうなる

れでかまいません。ただし，magic_dataの前でmagic_seekを忘れないようにしてください。

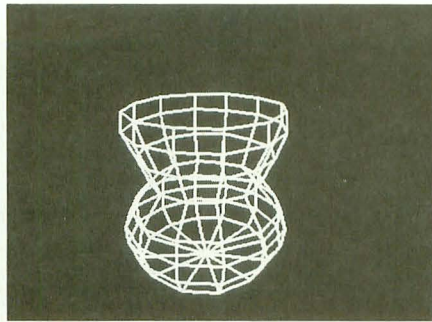
てなわけで

いかがだったでしょうか。基本的なことをダッシュで説明してきましたが，結構簡単でしょ？

あとはこれらの合わせ技ですから皆さんでいろいろやってみて，ゲームやフライトシミュレータなんか作っちゃってみてください。ただモデリングするのって結構たいへんですよね。方眼紙に三面図書いたりして。そこで簡易モデラーを作ってみました。なにが「簡易」かっつーと，ズバリ回転体しか作れない！　あんまり役に立たないかもしれませんが，使い方も簡単で見てるだけでも結構おもしろいのでぜひ使ってみてください。

回転体モデラーの使い方

実行するとマウスカーソルが出てきます。そんでもって画面の真ん中当たりに赤い線が引かれて右下にマウスカーソルの座標が出ます。この赤い線はX軸(すなわちY＝0)です。じゃあY軸は？　というと実は画面の一番左端がそうです。ここで勘のいい人ならもうわかったでしょう。マウスで画面に描いた図形をY軸を中心にぐるっと回してしまおうというのです。

試しになにか描いてみてください。左クリックで始点終点，右クリックでラバーバンドを切ります。点を打ちたいときは始点だけを指定して右クリックしてください。描けたらリターンキーを押します。すると「回転分割数？」とか聞いてきますので，ここではなにも考えずに12と入力しましょう。しばらくすると「おおっ！　これは!?」ってほどでもないですね。よーするにさっきの落書きを12回で1回転(つまり30度ごとだな）させた3Dデータを生成して表示するわけです。マウスの左右ボタンで大きさも変更できます。

なにかキーを押すとポイント数，ライン数，バッファ使用量が示され「CONTINUE?」と聞いてきます。続きを描きたいときはY・リターンとしてください。もういいというときはただリターンキーを押してください。すると今度はセーブするかどうか聞いてきますのでYかNを押して，セーブするときはファイルネームを入力してください。バイナリセーブされます。

アルゴリズム？

こういったモデラーを作るときに面倒なのは重複点や重複線を削除しなければならないことです。今回は重複点は削除するようにしていますが，重複線は無視しています。だって，わざわざそんなことする人はいないでしょう。べつにやったってどうってことないけど。それから回転させるとこはちょっとややこしいかもしれません。いまXY平面上に頂点がn個，線分がm本あったとしましょう（つまりマウスで描いた頂点と線分の数がそれぞれn,m)。そんでもってi番目とj番目の点が線分でつながれていたとしましょう。

ここで，たとえば30度ごとで回転させるとします。XY平面上の頂点はそのまま0～n－1番の頂点になります。次，30度回転させたn個の頂点は順番にn～2n－1番の頂点になります。その次，……となって，最後には330度回転させた頂点は11n～12n－1番の頂点になります。するとi番目とj番目の点は始めつながっていたので回転後もつながっているはずです。そうするとそれらの点を回転させたi＋n番目とj＋n番目の点もつながっているということになります。つまり「i番目とj番目がつながっていれば，i＋kn番目とj＋kn番目もつながっているんだよーん：kは自然数」の法則が成り立ってしまいます。

さらにi番目の点とi＋n番目の点は隣同士なのでつながっているべきで，またi＋n番目とi＋2n番目の点も同様です。というわけで「i＋kn番目とi＋（k－1）n番目はつながっているんだよーん：kは0もしくは自然数」の法則までもが成り立ってしまいます。と，思われましたが実はこれらの法則はある条件の下でしか成立しません。

が～ん，いきなり法則が破られてしまった。ではその条件とは？　それは「特異点が存在しないとき」です。そ，その特異点とは？　なんのことはない「回転中心に点が存在しないとき」です。だって，回転中心の点は回転しても移動しないもんね。重複点を無視してもかまわないならば，なにも考える必要はありません。

しかし，そうもいかないので次のようにすることにしました。まず，回転させる前に回転中心（つまりX＝0だな）にある点を先頭に移します。そして回転中心以外の点だけを回転させて番号をつけます。回転中心にあった点の個数をcとすると，さっきの法則は次のように書き換えることができます。

●第1法則

「i（≧c）番目とj（≧c）番目がつながっていれば，i＋k（n－c）番目とj＋k（n－c）番目もつながっているんだよーん：kは自然数」

●第2法則

「i≧cであればi＋k(n－c)番目とi＋(k－1)（n-c）番目はつながっているんだよーん：kは0もしくは自然数」

これで特異点が存在する場合も一般性を失わない。さらに，

●第3法則

「i（≧c）番目とj（<c）番目がつながっていれば，i＋k（n－c）番目とj番目もつながっているんだよーん：kは自然数」

よくわかんない人は無理に理解しなくていいです。脳味噌溶けちゃうかもしんないから（どこがアルゴリズムだ)。

最後に

このモデラーは回転体しか作れないといいましたが，実はそうではないのです。じ，じつは三角錐や直方体も作れるんです。とまあ冗談はこれくらいにして，なかなか遊べたでしょう？　モデラーとしてはあまり使えないけど。MAGIC.FNCのおかげでこういったことも，ずいぶん簡単になったことですし，そのうち誰かがまともなモデラーを作ってくれることを期待しましょう（と，伏線を張っておいて～)。それでは健闘を祈ります。

リスト6

```
  10 /*
  20 /*ＭＡＧＩＣ回転体モデラー
  30 /*          Programmed by T.Miki
  40 /*
  50 int i,j,k,x,y,l,r,c,xx,yy,lx,ly,no,by,w,z=160
  60 int p1,p2,center,offset,use_buf
  70 str ans,fn
  80 dim int buf1(4096)          /*ポイント
  90 dim int buf2(4096)          /*ライン
 100 dim int obj(8096)           /*３Ｄバッファ
 110 repeat
 120 magic_screen(1):console 0,31,0
 130 screen 1,2,1,1
 140 console ,,0
 150 vpage(3):apage(0)
 160 mouse(4):mouse(1)
 170 msarea(0,0,511,511)
 180 magic_flush()
 190 magic_init()
 200 magic_wipe()
 210 line(0,256,511,256,56)
 220 for i=0 to p2/2-1
 230  x=buf1(buf2(i*2)*2):y=buf1(buf2(i*2)*2+1)+256
 240  xx=buf1(buf2(i*2+1)*2):yy=buf1(buf2(i*2+1)*2+1)+256
 250  line(x,y,xx,yy,192)          /*既に描いたラインを表示
 260 next
 270 for i=0 to p1/2-1
 280  pset(buf1(2*i),buf1(2*i+1)+256,192)
 290  locate 48,30,0:print "X";buf1(2*i),"Y";-buf1(2*i+1);" "
 300 next
 310 magic_mode(1)            /*ＸＯＲモード
 320 center=0
 330 repeat           /*モデリング部のループ
 340 mspos(x,y)
 350 b=msstat(i,j,l,r)
 360 locate 48,30,0:print "X";x,"Y";256-y;" ";
 370 if c=0 then {
 380  if l=-1 then {
 390   no=point_search(x,y,p1)
 400   if no=p1 then {
 410    buf1(p1)=x:buf1(p1+1)=y-256
 420    p1=p1+2
 430    pset(x,y,255-192)
 440    print
 450   }
 460   buf2(p2)=no/2
 470   p2=p2+1
 480   lx=x:ly=y
 490   xx=x:yy=y
 500   c=1
 510  }
 520 } else {
 530  if x<>xx or y<>yy then {
 540   magic_line(lx,ly,xx,yy,15)
 550   magic_line(lx,ly,x,y,15)
 560   xx=x:yy=y
 570  }
 580  if l=-1 then {
 590   no=point_search(x,y,p1)
 600   if no=p1 then {
 610    buf1(p1)=x:buf1(p1+1)=y-256
 620    p1=p1+2
 630    print
 640   }
 650   buf2(p2)=no/2:buf2(p2+1)=no/2
 660   p2=p2+2
 670   magic_mode(2)
 680   magic_line(lx,ly,x,y,192)
 690   magic_mode(1)
 700   pset(x,y,255-192)
 710   lx=x:ly=y
 720  }
 730  if r=-1 then {
 740   p2=p2-1
 750   c=0
 760   magic_line(lx,ly,x,y,15)
 770  }
 780 }
 790 while l<>0         /*ボタンが離されるのを待つ
 800 msstat(i,j,l,r)
 810 endwhile
 820 until inkey$(0)<>""
 830 print
 840 if c=1 then {
 850  c=0
 860  p2=p2-1
 870  magic_line(lx,ly,x,y,15)
 880 }
 890 /*回転中心にあるポイントをバッファの先頭に移す
 900 for i=0 to p1-1
 910  if buf1(i)=0 then {
 920   center=center+1
 930   by=buf1(i+1)
 940   for j=0 to i-1
 950    buf1(i-j+1)=buf1(i-j-1)
 960   next
 970   buf1(0)=0:buf1(1)=by
 980   for j=0 to p2-1
 990    no=buf2(j)
1000     if no<i/2 then buf2(j)=no+1
1010     if no=i/2 then buf2(j)=0
1020    next
1030   }
1040   i=i+1
1050 next
1060 mouse(2)
1070 repeat
1080 input "回転分割数";w
1090 until w>2
1100 obj(0)=(p1/2-center)*w+center      /*ポイント数
1110 j=1
1120 /*回転中心にあるポイントを３Ｄバッファに書き込む
1130 for i=0 to center-1
1140  obj(j)=buf1(i*2)
1150  obj(j+1)=buf1(i*2+1)
1160  obj(j+2)=0
1170  j=j+3
1180 next
1190 /*回転中心以外のポイントを３Ｄバッファに書き込む
1200 for i=0 to w-1
1210  for k=center to p1/2-1
1220   obj(j)=buf1(k*2)*cos(pi(2)*i/w)
1230   obj(j+1)=buf1(k*2+1)
1240   obj(j+2)=buf1(k*2)*sin(pi(2)*i/w)
1250   j=j+3
1260  next
1270 next
1280 obj(j)=(p2/2+p1/2-center)*w        /*ライン数
1290 j=j+1
1300 /*経線（？）の連鎖を３Ｄバッファに書き込む
1310 for i=0 to w-1
1320  for k=0 to p2-1
1330   no=buf2(k)
1340   if no>center-1 then no=no+i*(p1/2-center)
1350   obj(j)=no
1360   j=j+1
1370  next
1380 next
1390 /*緯線（？）の連鎖を３Ｄバッファに書き込む
1400 for i=0 to w-1
1410  for k=center to p1/2-1
1420   obj(j)=k+i*(p1/2-center)
1430   if i=w-1 then obj(j+1)=k else obj(j+1)=k+(i+1)*(p1/2-
cen ter)
1440   j=j+2
1450  next
1460 next
1470 offset=magic_putbuf(1,obj)          /*３Ｄデータを渡す
1480 magic_seek(1,offset,0)
1490 magic_wipe():cls
1500 magic_mode(2)
1510 magic_color(252)
1520 for i=0 to 8
1530  magic_para(i,0)
1540 next
1550 repeat
1560 magic_data(1)
1570 magic_para(1,10)   /*Ｘ座標
1580 magic_para(2,z)    /*Ｚ座標
1590 magic_para(6,j)    /*ＨＥＡＤ
1600 magic_para(7,330)  /*ＰＩＴＣＨ
1610 magic_pers()
1620 magic_disp()
1630 j=j+4
1640 msstat(i,k,l,r)
1650 if l=-1 and z>0 then z=z-10
1660 if r=-1 and z<800 then z=z+10
1670 until inkey$(0)<>""
1680 console 0,32,0
1690 print "ポイント数";obj(0)
1700 print "ライン数";obj(obj(0)*3+1)
1710 use_buf=(obj(0)*3+obj(obj(0)*3+1)*2+2)*2
1720 print "使用バッファ数";use_buf;"バイト"
1730 ans=" "
1740 input "ＣＯＮＴＩＮＵＥ(Y/Else)";ans
1750 until ans<>"y" and ans<>"Y"          /*ふりだしに戻る
1760 repeat
1770 print "ＳＡＶＥしますか(Y/N)";:ans=inkey$:print ans
1780 if ans="y" or ans="Y" then {
1790  input "filename";fn
1800  magic_save(1,fn,offset,use_buf)  /*バイナリセーブ
1810  break
1820 }
1830 until ans="n" or ans="N"
1840 end
1850 /*
1860 /*ポイントバッファに（Ｘ，Ｙ）があるか調べる
1870 /*
1880 func point_search(x,y,p1)
1890 int i
1900 for i=0 to p1-1
1910  if buf1(i)=x and buf1(i+1)=y-256 then break
1920  i=i+1
1930 next
1940 return(i)
1950 endfunc
```

付録 MAGIC.FNCとMAGICダンプリストの入力法

編集部

7月号で発表したMAGIC.FNCのデバッグ版をダンプリストで掲載します。ついでにMAGIC本体(ダンプリスト)の入力も含めて簡単に解説します。これからはX-BASICからMAGICアプリケーションを作成することも可能です。

7月号で発表したMAGIC.FNCではソースリストしか掲載できなかったため,入力を躊躇してしまった人も多いと思います。さらに最近はプリントアウトされたリストの印刷品質が下がってきており,可読性の低いプログラムで読者の皆さんにはたいへんご迷惑をおかけしてしまいました。

今回は入力ミスの出にくいダンプリストのかたちで掲載します。存分にMAGIC.FNCを活用してください。また,今回の特集ではMAGIC本体がダンプリストのかたちで掲載されています。同様な操作で入力してください。

なお,ダンプリストは圧縮された状態で掲載されていますので,実際の使用時にはLH.X (Oh!X1991年5月号ほかに収録) が必要になります。また,予定していましたC言語用ライブラリは後日発表します。

入力ツールの使い方

まず,入力ツールですが,コンパイルしたものが1990年6月号付録ディスクに収録されています。お持ちでない方はリスト2をX-BASICから入力してください (できるだけコンパイルしてご使用ください)。

マシン語入力ツールを起動すると,最初に新規作成のファイルかどうかを聞いてきます。初めて操作するときはYキー,そうでないときはそれ以外のキーを押してください。次にファイル名を入力します。この場合は,MAGIC.LZH, MAGICFNC.LZHと指定してください。

次に入力です。X-BASIC上で動作させている方以外はCキーを押してCRCチェックバイト (各ブロックの右下の数字) を表示させてください。Eキーでエディットモードに入ります (CAPSキーはOFFにしておいてください)。ひたすらダンプリストどおりに打ち込み,1ブロック入力するごとに縦横のチェックサム (縦横のデータを合計したもの) 部分があっているかを確認します。入力を中断するときはESCキーを押してSでセーブできます。ファイルサイズを聞いてきますが,単にリターンキーを押すだけでもかまいません。すべてを入力し終えたときに限り,ファイルサイズ,

MAGIC.LZH　4590
MAGICFNC.LZH　2679

でセーブしてください。

あとはLH.Xを用意し,

LH -E MAGIC
LH -E MAGICFNC

で必要なファイルが作成されます。

リスト1

```
0000  1F C0 2D 6C 68 31 2D 55  : 93
0008  0A 00 00 14 15 00 00 E6  : 19
0010  70 F1 16 20 00 09 6D 61  : 6E
0018  67 69 63 2E 66 6E 63 17  : AF
0020  D6 EA 78 71 92 80 0C E0  : A7
0028  F6 1D FC 77 FA FE 35 14  : C7
0030  72 DB 6E D7 25 43 34 60  : 8E
0038  0C C9 05 40 0C 8A CB 33  : AE
0040  4F 91 84 DE AA 74 7B D6  : B1
0048  05 9C F3 CF DF 79 DA EE  : 83
0050  BD B3 CC 56 D5 81 EF 79  : 50
0058  B0 B3 96 35 A0 B3 05 E0  : 66
0060  29 34 DB CE 2C 40 3A C6  : 72
0068  FB 91 1C AD 14 72 FF AF  : 89
0070  8C 61 E8 8F DB 2C BD B8  : E0
0078  C5 D6 77 35 CE 5F 37 6D  : 18
-------------------------------------
SUM:  80 54 BC 44 87 51 B3 F1  42A1

0080  C0 94 15 2E A8 0C F9 4B  : 8F
0088  C9 5A 97 05 8C 75 30 DC  : CC
0090  32 60 D4 ED 83 55 B0 1B  : F6
0098  17 A3 AB 54 FA 2A 70 4E  : 9B
00A0  6D FC 53 5C 48 B5 50 CE  : 33
00A8  D5 CF F3 9D BB 40 3D C2  : 2E
00B0  E6 85 E8 52 0E D0 A0 76  : 99
00B8  57 1F B2 43 D1 A8 94 08  : 80
00C0  D4 D0 14 16 53 17 49 BB  : 3C
00C8  31 7C 42 CA 4F F8 92 61  : F3
00D0  C5 92 07 5E 53 30 DE C5  : E2
00D8  48 F3 0E 5A 7B B8 A8 96  : 14
00E0  6E BC A7 EC 20 A7 62 80  : 66
00E8  3B F6 99 21 31 E2 56 51  : A5
00F0  E2 77 51 04 A4 2B 35 89  : 3B
00F8  0F AE 85 59 5D 44 E8 D7  : FB
-------------------------------------
SUM:  FD 08 8C 04 55 5C 40 46  C8F2

0100  12 1F CC 0F 03 7F FC 0D  : 97
0108  9F E4 75 C4 31 D0 96 66  : B9
0110  0B 7F C7 F7 88 15 B0 03  : 98
0118  CC 0F 4B 7C A0 07 E6 E1  : 10
0120  40 FC B8 49 B8 25 BA D9  : AD
0128  D9 F4 4A 9D C7 45 BF 38  : B7
0130  83 7A 61 4B 54 C3 18 47  : 1F
0138  F6 21 03 ED B6 4B 4E 3F  : 95
0140  08 B5 B6 EF 7F 5F 22 5B  : BD
0148  B0 C8 84 B4 70 6D 91 2D  : 4B
0150  47 D2 12 DE 2E 1B 20 68  : DA
0158  56 07 A6 6D 25 A0 B6 B9  : A4
0160  BE 15 02 C9 B2 A6 54 B7  : 01
0168  06 B6 74 B7 54 13 4D 2D  : C8
0170  A8 3F D2 DE E3 6B 1F F3  : F7
0178  D8 1E 39 D0 5A D0 EC 91  : A6
-------------------------------------
SUM:  B3 9A 2C 80 6A 5E 3C FF  000F

0180  28 59 6A 70 B0 38 CD F2  : 02
0188  D5 65 60 D7 67 92 D7 DC  : 1D
0190  44 FE 2D 50 36 5A E7 A2  : D8
0198  33 4B 5C 12 6C 7E 5A B0  : E0
01A0  38 DB E0 B5 9D 58 7C 44  : 5D
01A8  07 9F F1 4B 26 44 AF 8B  : 86
01B0  5E C9 08 7A 56 EA EA CD  : A0
01B8  BF A6 9D FC 07 DA DE E7  : A4
01C0  DA 6F EC 02 F4 3B 6A 2B  : FB
01C8  FA A7 49 D6 F3 BF 07 8A  : 03
01D0  0D AA 3C 36 79 A8 A5 05  : F4
01D8  AF C0 B7 9B AF FF 3C 1A  : C5
01E0  ED 4E B0 3A 98 05 E0 1E  : C0
01E8  BF 64 94 07 E9 B5 4A E0  : 86
01F0  86 8D C7 A9 A0 57 F2 D6  : 42
01F8  05 99 CC D1 C3 81 53 56  : 28
-------------------------------------
SUM:  97 48 C8 83 CC 35 99 A1  4DBA

0200  7A F6 BE AA D6 3D C7 C7  : 79
0208  BD 6D 9D FC 35 9E F3 2B  : B4
0210  5A 10 35 74 B3 D9 B6 EC  : 41
0218  83 44 B2 DD D5 92 B0 9B  : 08
0220  6A 87 C9 BC 9C C4 12 6E  : 56
0228  55 93 76 DF F2 AD BF E5  : 80
0230  BE 6F F6 44 D3 D2 11 B0  : CD
0238  16 46 FE 54 48 DA 86 2D  : 83
0240  3A 75 63 2C 46 9C A1 A1  : 62
0248  E0 B3 35 5B 3D C6 B4 C6  : A0
0250  CF 52 39 1C 72 F7 C3 A0  : 42
0258  8E F3 C7 6A 7F 94 E6 A4  : 4F
0260  0F BE B6 C7 76 59 F0 AC  : B5
0268  BB 56 B0 39 3E 4F D9 D0  : 30
0270  54 B7 F3 EC DF 84 0B 7F  : D7
0278  BE C3 3F 88 B5 C9 4B 53  : 64
-------------------------------------
SUM:  FA 81 A5 AB F8 45 A5 A2  0FFC

0280  95 C6 39 94 4D 73 60 F0  : 38
0288  7C 9E D4 1C 21 8F F8 5D  : 0F
0290  3F 94 A5 ED E7 CD 62 A5  : 20
0298  44 9D E1 ED DB BD 8B A3  : 75
02A0  B5 7E 74 3C 2A A9 EF 95  : 3A
02A8  5F 0B F3 DE 13 7F 84 89  : DA
02B0  5D 91 E2 23 D3 C9 83 DB  : ED
02B8  18 A5 5D 31 4B 16 E9 3E  : D3
02C0  44 F7 7C 0F 72 42 32 83  : 2F
02C8  15 E5 5E C1 E0 7E 7A 3B  : 2C
02D0  61 4B E0 0F 12 CE AE 93  : BC
02D8  C8 FB 55 78 F4 95 CA 5E  : 41
02E0  99 4F 66 F4 B4 37 73 C5  : 65
02E8  FE 25 22 8D 3B D6 CD 35  : E5
02F0  3E 80 C7 50 D8 BC CE 73  : AA
02F8  C0 E2 D2 1F A8 D9 0D 57  : 78
-------------------------------------
SUM:  34 4C 69 3F 52 58 63 3F  0ED0

0300  2C 3B B7 6B 6B 30 EF 16  : 29
0308  C7 86 41 F9 96 E7 4F 78  : CB
0310  17 21 CC 39 F2 53 D3 41  : 96
0318  9C 45 60 B3 EC B7 63 63  : 5D
0320  56 3B 13 F3 C1 53 CB 55  : CB
0328  5F B3 7E 15 FF 73 9F 15  : CB
0330  FC 44 A5 D3 61 66 FB A7  : 21
0338  EB 3A E0 0C 87 4A A3 59  : DE
0340  0D E6 48 AF 0D 22 CC 1A  : FF
0348  D2 E2 95 A6 EC 9A 01 13  : 89
0350  FC EE EF FB 3D B8 5A C2  : E5
0358  9C FE B3 73 87 2D F0 87  : EB
0360  F2 E6 F0 5B 84 B6 43 D2  : 72
0368  03 3D 20 D1 A7 95 17 F6  : 7A
0370  5B E9 9F A2 71 C6 7C B7  : EF
0378  80 E6 DB 64 4B FF 6D 7D  : D9
-------------------------------------
SUM:  89 39 43 2C 2B 48 D6 0E  BBB1
```

```
0380   7D 8E 42 AA 8A 1E 29 6F   : 37
0388   5D 1D CA E6 E7 0E 79 8D   : 25
0390   DE E1 4C 69 C3 1D FC 2D   : 7D
0398   7F 37 7A 65 6C CF 91 C5   : 26
03A0   EC C2 C3 DC 97 DF 13 A5   : 7B
03A8   AC 13 B7 1C 2B EA BF 7D   : E3
03B0   58 19 3D E4 A2 3D F4 D5   : 3A
03B8   49 DF 1B 75 6C A7 64 F8   : 27
03C0   35 2B 6D BF F0 05 11 57   : E9
03C8   3F 63 53 97 E1 C5 7F 43   : F4
03D0   73 8E 8E 36 38 92 A0 EB   : 1A
03D8   B8 3C 19 1B 4B E2 30 72   : F7
03E0   DD 3D D5 F7 2D B0 D9 C1   : 5D
03E8   35 1A CF 7D 20 B1 D3 04   : 43
03F0   A3 9B E9 83 CC C2 E8 D5   : F5
03F8   19 A0 78 8F 39 A9 84 71   : 97
-------------------------------------
SUM:   DD 7A 10 DC 16 CF D1 DF   863C

0400   8A C0 C7 4A B4 C2 0B 51   : 2D
0408   52 B9 C4 F9 92 2A BE 07   : 49
0410   59 AB 3E 8E F9 51 63 27   : A4
0418   78 E5 57 5D CD 55 64 D0   : 67
0420   14 FE CF EC 8D A6 47 48   : 8F
0428   F2 A0 B9 91 73 09 EB 78   : BB
0430   F2 89 50 B3 B3 2A 9C FC   : F3
0438   D2 1C FD 6D 43 1B 01 56   : 0D
0440   23 4A B5 21 B2 44 D2 69   : 74
0448   9E 2D 89 2D 66 A5 FE CE   : 58
0450   E1 14 A0 85 61 91 D9 DC   : C1
0458   1D A6 F2 91 E7 CB 21 A7   : C0
0460   1D 06 C4 F1 AB 0F E1 AA   : 1D
0468   3B 03 F9 AA 5E 63 9E 3E   : 7E
0470   EE 6F 8E 9C 51 81 50 F9   : A2
0478   1A 41 A9 5B E0 53 72 73   : 77
-------------------------------------
SUM:   96 36 B9 C1 9C 11 6A 6F   A36E

0480   66 F9 38 48 D6 70 C7 90   : 7C
0488   DE D3 E4 1F B1 BB 68 C8   : 50
0490   C9 FD 30 F1 20 8C 38 87   : 52
0498   CE 30 C7 0E 03 96 7F 30   : 1B
04A0   E1 38 C1 6C E8 32 9F 89   : 88
04A8   97 A2 F6 52 6F 06 F1 4F   : 36
04B0   C5 AA 5E 21 B3 08 DD 27   : AD
04B8   64 DD C9 76 62 A8 94 BE   : DC
04C0   77 62 2E 4D C6 52 3E CD   : 77
04C8   BE 49 F4 A5 5D 8A E1 6D   : D5
04D0   5D 91 6F BB 7D AD 8F 90   : 61
04D8   3B 26 F5 2C 8C 4B 17 01   : 71
04E0   33 C8 62 27 D3 64 91 CE   : 1A
04E8   5C F8 86 F9 FD 95 9E 8B   : 8E
04F0   E6 AE 9A 1D 34 2A 11 CA   : 84
04F8   30 0E D4 B9 AB 45 85 46   : 86
-------------------------------------
SUM:   EE 38 CD 8A F1 71 71 00   3ACA

0500   BD A2 D0 FE 38 10 64 0B   : E4
0508   E3 5B B5 07 40 E6 5F 1B   : 9A
0510   0B 1A 91 53 FF C2 3C D7   : DD
0518   F3 C6 CF FE 55 11 44 5D   : 8D
0520   14 B3 0F 6D 48 BE F0 3C   : 75
0528   38 DE CC CF E3 96 54 C7   : 45
0530   C2 89 8A F3 42 DB 6D BF   : 11
0538   31 E8 E5 82 D3 29 A1 3B   : 58
0540   AB DB A7 58 2B 5A FD A1   : A8
0548   24 4A EA 06 F6 1E E1 4E   : A1
0550   A6 1C A3 1A E5 7C A1 B8   : 39
0558   CA A5 41 FA 2B E3 9C E7   : 3B
0560   44 A7 5C CC D1 B9 1E 81   : 3C
0568   E7 F4 03 E1 76 EA 23 BC   : FE
0570   D9 F5 AE 78 CE 55 35 6E   : BA
0578   BB 70 51 6E BF 92 EA BB   : E0
-------------------------------------
SUM:   DB C5 02 0C 11 82 10 4B   19E3

0580   E8 1E 82 EE 2B 1D 49 B5   : BC
0588   F5 5E DB 1F 57 AE 02 97   : EB
0590   65 2F 69 B9 0F DC 5D B9   : B7
0598   71 75 3C B8 A3 3D 5A 07   : 1B
05A0   9D D6 66 3E 5C BA D1 AD   : AB
05A8   CE AC FC F0 DC A5 AA 40   : D1
05B0   02 C9 57 86 33 C4 96 8E   : C3
05B8   EF D8 0F 25 E0 A9 8A 69   : 77
05C0   6F 0D 2D E2 83 F0 90 AA   : 38
05C8   46 0A 75 ED A0 F0 20 C6   : 28
05D0   DB A5 9C 38 3B DA 41 4D   : F7
05D8   63 73 18 CB FD 66 A4 7A   : 3A
05E0   52 1C A6 69 97 4C D8 D8   : 10
05E8   E7 5D F8 DF 60 E3 77 EF   : C4
05F0   B5 80 FE 1A B6 65 43 02   : AD
05F8   77 02 88 A1 2A 36 2D B0   : DF
-------------------------------------
SUM:   67 6D 44 2C B1 9A F1 A0   82A8

0600   9C C1 D9 7E 70 83 BF 2B   : 91
0608   AD 9F 95 D8 D6 D1 EA DE   : 28
0610   3C 9A 7A 99 B6 FA 3D 19   : EF
0618   E3 08 AC 27 E5 3F 34 38   : 4E
0620   8A 2E 6E DA 5A 72 CD 0D   : A6
0628   D4 B6 F7 9C 9E C0 73 62   : 50
0630   F4 30 EB 54 3B 17 A0 B4   : 09
0638   75 CD B4 5E C8 47 8E D7   : C8
0640   3E 25 93 EA BE 21 97 1F   : 75
0648   52 BB F1 3A A7 22 9A E6   : 81
0650   8C 4A F6 23 30 B1 41 56   : 67
0658   5A C5 E9 BB AA 76 D9 57   : 13
0660   89 C9 36 49 27 E2 7A 4B   : 9F
0668   93 ED B6 CA B0 91 D8 A4   : BD
0670   B1 AC F3 EB 4D 82 0A A8   : BC
0678   D7 0B D5 72 EA 95 4A F4   : E6
-------------------------------------
SUM:   49 3F AF B0 29 11 79 91   E8C1

0680   E0 5E 98 AF 76 CF 3B 68   : 6D
0688   BE 2B B4 AA 80 9B 92 7F   : 73
0690   F0 3E A5 6B BB 5B C6 2B   : 45
0698   F2 07 8C 78 F0 6C 92 CE   : B9
06A0   13 4E 7C 72 C4 79 8E CB   : E5
06A8   CF 7F 41 E0 5B A7 85 0E   : 04
06B0   0B 0C E0 BF 31 51 4E 5D   : E3
06B8   E6 6B 6A 0A 6F F1 CE 5F   : 52
06C0   E2 C7 86 C8 2B 0C CA 2E   : 26
06C8   04 C9 88 73 D7 E1 55 D0   : A5
06D0   27 29 F2 04 D4 C2 81 2C   : 89
06D8   32 E0 99 3C 43 01 39 D1   : 35
06E0   20 4C 70 E5 C7 C8 CA 00   : 1A
06E8   DF 2F 2C 50 E0 05 EA 88   : E1
06F0   60 BD 11 0E 17 A5 1B 05   : 18
06F8   F7 15 37 5D A0 26 04 E0   : 4A
-------------------------------------
SUM:   E8 F8 01 72 D7 DB 00 DD   BC84

0700   95 50 B6 16 88 8B 42 86   : 8C
0708   44 22 FB 06 7C 99 25 09   : AA
0710   4E 4D 81 29 C8 00 98 AA   : 4F
0718   44 C4 13 84 FD FC 12 88   : 32
0720   F9 15 A5 0A 6A 7F 12 BE   : 76
0728   BD E9 83 C0 A7 DD 2D 56   : F0
0730   CD 99 EF 4B EE 06 D6 FF   : 69
0738   38 E8 6A 37 FC 82 F4 09   : 3C
0740   86 07 9B 40 F9 CE 9C 3C   : 07
0748   43 0B 6E E6 4E A7 E8 7D   : FC
0750   E2 34 37 BC 17 8D 63 24   : 34
0758   E1 8D 61 0F C0 C8 81 78   : 5F
0760   48 E0 BF F0 D0 50 A1 AC   : 44
0768   43 43 05 17 59 F3 2A 0A   : 22
0770   29 C2 0A 11 54 15 60 CA   : 99
0778   D4 F1 36 8C A0 F3 D6 9B   : 8B
-------------------------------------
SUM:   3A AB 6B AA FF 19 83 4D   F66E

0780   0C 3D BC 3A AF 88 0C 81   : 03
0788   6F CC 74 78 2E 39 0F CD   : 6A
0790   7E 6A AB 2A AB E2 8E 20   : F8
0798   5E 8B 72 93 6D 6A 99 59   : B7
07A0   0C 62 83 14 FB FB 03 3C   : 3A
07A8   8E 0D FC C4 8F 9E 6F 14   : 0B
07B0   09 85 7B 9A E2 40 83 91   : D9
07B8   FD 46 78 3B E3 06 01 9B   : 7B
07C0   A0 C0 01 19 81 05 F6 7F   : 75
07C8   07 EB 1F 22 8D 01 AC D0   : 3D
07D0   1B D6 80 F6 36 FD F7 F2   : 83
07D8   B4 B7 C3 40 71 06 80 CA   : 2F
07E0   F4 07 B7 82 C7 02 E8 0E   : F3
07E8   AC D0 1C 9B 80 7D 97 FC   : C3
07F0   3E C6 DE D7 9B 80 65 7E   : B7
07F8   01 C7 3C 03 AB 38 07 B6   : A7
-------------------------------------
SUM:   4C D4 0F 84 86 2C 3C 8C   8DCF

0800   E0 1C 2B 80 67 3E 01 F7   : 44
0808   E3 FF 7B FE 01 D1 BA 1A   : 01
0810   E9 E0 1F E7 00 E4 5E 03   : 14
0818   C4 F0 19 BA FB 86 01 F7   : 00
0820   13 9F C0 7A 41 E4 04 E4   : F9
0828   58 2F 64 1B C2 04 EB 0D   : C4
0830   EF A7 82 C4 0D D2 17 E5   : B7
0838   C1 BE E6 C4 00 FF 2F 51   : A8
0840   80 A6 82 5C 5A F5 AF 9C   : 9E
0848   2F 6E 23 50 3F FB 02 14   : 60
0850   48 9B B6 FC 03 88 DD F8   : F5
0858   2F C0 38 EA EC CC BC 03   : 88
0860   ED C5 9B C9 4C AF 58 3D   : A6
0868   7E 93 7B 7A C1 2B D4 8E   : 54
0870   1B 2F 01 BD BC 07 AB E0   : 56
0878   31 67 80 C5 BE 03 F5 43   : D6
-------------------------------------
SUM:   68 7B 94 93 82 5A 65 CB   8698

0880   93 28 BB C7 87 76 A8 4F   : 31
0888   F0 3F 8D E8 F6 57 74 FC   : 61
0890   77 91 E3 3A D1 05 DA C6   : 9B
0898   B8 05 BE 80 1C 34 00 65   : B0
08A0   A3 64 FD E9 69 AB F5 72   : 68
08A8   23 AF 8D 25 AB 48 CD C9   : 0D
08B0   8B 9B CB 31 3C 6E F5 7F   : 40
08B8   9A BE 22 5A A5 76 E8 B9   : 90
08C0   D0 11 83 55 D2 00 2F 32   : EC
08C8   50 BC 6D 22 B9 FE 99 8F   : 7A
08D0   4A 40 77 A3 E6 8B A3 0A   : C2
08D8   02 FB 53 D3 41 17 66 21   : 02
08E0   74 9D AF A6 4D 46 05 02   : 00
08E8   1F 8C D5 6B BA C6 3F 8D   : 37
08F0   37 CA B9 BC EB E0 48 88   : 11
08F8   F8 9C DA 71 0A 9F A6 08   : 36
-------------------------------------
SUM:   CB 00 31 2D 0D 08 98 F4   7841

0900   25 52 17 CA F5 84 6C D8   : 15
0908   4C 73 5F 47 EA 03 FC F2   : 40
0910   85 AA BE 0C 99 E0 09 5E   : D9
0918   10 03 63 A6 8F 47 57 00   : 49
0920   0B 79 ED F0 30 05 C1 B6   : 0D
0928   0A F6 B9 BD 90 BE 5F 7A   : 9D
0930   74 2B B0 AF 43 17 CE 7A   : A0
0938   57 77 4E 70 85 80 AE A9   : E8
0940   75 8A 29 2D E6 29 7C 86   : 66
0948   F3 99 A0 52 BB 2C 2C 20   : B1
0950   99 51 22 A3 FD B3 9C 3B   : 36
0958   70 81 07 F7 88 AF B9 24   : 03
0960   9C 5D FD 25 06 9F C7 79   : 00
0968   C2 04 84 A7 6D 04 4C 5B   : 09
0970   E4 AC D5 77 C8 DC 0F 42   : D1
0978   15 BB 17 8A 82 D9 48 4A   : 5E
-------------------------------------
SUM:   AE 40 9A 75 72 17 CB E0   D1A4

0980   AC 6C 3E E0 9B 87 CE 46   : 6C
0988   EC 33 D6 2E 26 49 3C 4F   : 1D
0990   2C B1 B8 72 0F 64 6E BB   : A3
0998   A9 9B 91 2F AA 3E 51 5D   : 9A
09A0   FD 23 DC 03 67 23 14 B0   : 4D
09A8   15 40 89 E4 78 E7 98 B6   : 6F
09B0   0B 8C 46 4D AF F2 21 FC   : E8
09B8   B7 93 56 F9 66 F6 02 3B   : 32
09C0   E1 11 66 C4 D0 03 ED 9C   : 78
09C8   CF 81 66 9C 03 10 2D E0   : 72
09D0   00 31 5F AA 2D A0 2D C9   : FD
09D8   82 60 07 C2 C0 2B 40 1B   : F1
09E0   EA EB 53 3C 7F 25 56 E9   : 47
09E8   62 48 26 37 C1 26 04 F2   : E4
09F0   74 02 7A 0F 20 26 78 13   : D0
09F8   77 F0 49 B2 41 26 7A 41   : 84
-------------------------------------
SUM:   AA B5 CC DC CF D9 6B D9   BC87

0A00   2D 65 82 4C DE 4E 98 F8   : 1C
0A08   24 D2 4C 3F D4 02 E2 80   : B9
0A10   36 68 0E DE 00 B4 79 60   : 17
0A18   BA 93 40 5F C0 2B EC 9C   : 5F
0A20   9D 30 21 76 0E 5C 20 17   : 05
0A28   F8 4E 64 01 BF AD 84 7E   : 19
0A30   20 3E D0 3D A0 6C 8C 0A   : 0D
0A38   F1 6B 98 00 B3 A9 A8 0C   : 04
0A40   DD 05 09 50 2C E5 A0 A7   : 93
0A48   E5 00 F5 D0 15 E6 AB 70   : C0
0A50   11 CC F0 0C 86 3F 21 B0   : 6F
0A58   82 0F FD 80 6E 40 17 B4   : 87
0A60   01 5D 00 BE 09 4A 05 8C   : 00
0A68   03 87 A1 2C 5B 39 A8 07   : 9A
0A70   4F 64 03 9E EA 80 00 00   : BE
0A78   00 00 00 00 00 00 00 00   : 00
-------------------------------------
SUM:   8F 81 98 B0 15 9A E7 2D   BB44
```

リスト2

```
 10 /* program macinto-c_pro68k(input,output);
 20 /* var
 30     char Dump(65535),A1
 40     int Num,Pointer=-8,Size,Size1,Data,Sum,Vsum(7)
 50     int Work(7),X,Y,F,M,CrcOn=1,EF=0
 60     str Hex,EditFile,Mode="r",Ascii,B1,Hyoji,Dam
 70 /* begin
 80 cls
 90 print "New file ( y or n )":B1=inkey$
100 if strlwr(B1)="y" then Mode="c"
110 input "Edit file := ";EditFile
120 Num=fopen(EditFile,Mode)
130 Size=fseek(Num,0,2)
140 fseek(Num,0,0)
150 if not Size=0 then{
160    fread(Dump,Size mod 65536,Num)}
170 fcloseall()
180 print EditFile,Size;"Byte (";hex$(Size);"H)" :print
190 print " 'T' = P a g e  U p        'P' = P r i n t  Ou
t"
200 print " 'G' = P a g e  D o w n    'C' = C R C ON/OFF"
210 print " 'E  = E d i t  M o d e    'Esc'= Command Mode
```

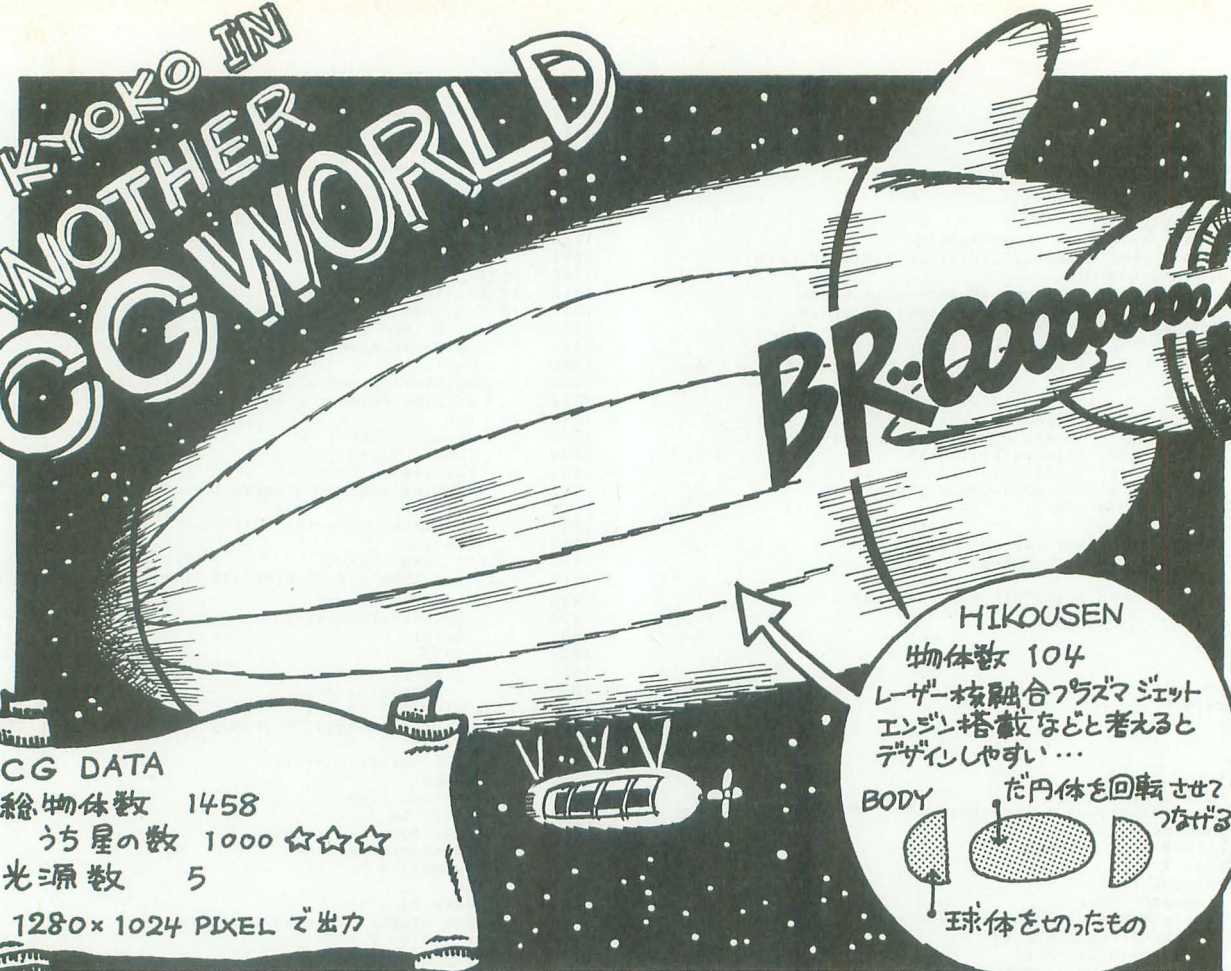
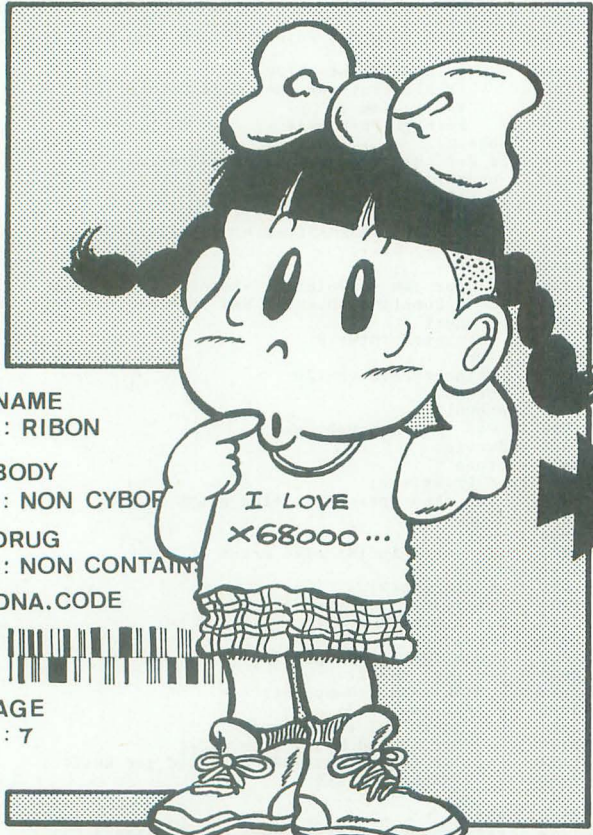

▶先日，日本橋の中古ソフト店であの「いなばの白兎」を29円で買った。カセットBASICからの2段ロード，うむをいわさぬゲーム内容，そして万全の2回セーブ。私はゲームソフトの原点を見た。
木下　達也(19)兵庫県

```
 220 print " 'S' = S a v e           '!' - Q u i t"
 230 /*
 240 locate 0,11
 250 repeat
 260    repeat
 270      Out()
 280    until Pointer > (Size)*abs(M)
 290    M=0: print"Command:";:B1=inkey$:B1=strlwr(B1)
 300    switch asc(B1)
 310      case  't' : Pointer=Pointer-128:break
 320      case  'g' : Pointer=Pointer+128:break
 330      case  'e' : Edit():break
 340      case  's' : Num=fopen(EditFile,"w")
 350                  input "ファイルサイズを入力してください
",Size1
 360                  if Size1<>0 then Size=Size1
 370                  fwrite(Dump,Size mod 65536,Num)
 380                  locate 0,29:print space$(45);
 390                  fcloseall():break
 400      case  'p' : M=1:Pointer=-8:break
 410      case  'c' : CrcOn=-CrcOn:break
 420      case  '!' : EF=1
 430    endswitch
 440    Pointer=Pointer-128
 450    if Pointer<-9 then Pointer=-8
 460    locate 0,8
 470    if M=1 then print"Hit Key":B1=inkey$
 480    locate 0,8:print"          "
 490 until EF=1
 500 end
 510 /*
 520 func Out()
 530 locate 0,10
 540 for i=0 to 7
 550   Vsum(i)=0
 560 next
 570 for i=0 to 15
 580   Pointer=Pointer+8
 590   Hex=string$(4-len(hex$(Pointer)),"0")+hex$(Pointer)
 600   Dam=inkey$(0)
 610   Pr(Hex+"  ")
 620 Ascii=""
 630   for j=0 to 7
 640     Data=Pointer+j
 650     Pr(string$(2-len(hex$(Dump(Data))),"0"))
 660     Pr(hex$(Dump(Data))+" ")
 670     Sum=Sum+Dump(Data)
 680     Vsum(j)=Vsum(j)+Dump(Data)
 690 A1=Dump(Data)
 700 if not isprint(A1) then A1=&H2E
 710     Ascii=Ascii+chr$(A1)
 720   next
 730   Pr(" : "+right$("0"+hex$(Sum),2)+"   "+Ascii)
 740   Prl()
 750   Sum=0
 760 next
 770 Pr(string$(35,"-"))
 780 Prl()
 790 Pr("SUM:  ")
 800 for i=0 to 7
 810   Pr(right$("0"+hex$(Vsum(i)),2)+" ")
 820 next
 830 if CrcOn=-1 then Pr(" "):Pr(Crc(Pointer))
 840 Prl():Prl()
 850 endfunc
 860 /* edit mode
 870 func Edit()
 880 Pointer=Pointer-120
 890 X=0:Y=0
 900 while 1
 910   for i=0 to 7
 920     Work(i)=Dump(Pointer+i)
 930   next
 940   while 1
 950   locate X/2+X+6,Y+10
 960   F=0
 970   repeat
 980     B1=inkey$
 990     switch asc(strlwr(B1))
1000      case 28:X=X+1:F=1
1010             if X=16 then X=0:F=2
1020             break
1030      case 29:X=X-1:F=1
1040             if X=-1 then X=15:F=3
1050             break
1060      case 30:F=3:break
1070      case 31:F=2:break
1080      case 13:F=4:break
1090      case 27:F=5:break
1100      case 127:F=6:break
1110      default
1120       if B1>="0" and B1<="9" then A1=asc(B1)-48:F=1
1130       if B1>="a" and B1<="f" then {A1=asc(B1)-87:F=1
1140                                   B1=chr$(asc(B1)-32)}
1150       if asc(B1)=12 then F=1:A1='a'-87:B1="A"
1160       if asc(B1)=47 then F=1:A1='b'-87:B1="B"
1170       if asc(B1)=42 then F=1:A1='c'-87:B1="C"
1180       if asc(B1)=45 then F=1:A1='d'-87:B1="D"
1190       if asc(B1)=43 then F=1:A1='e'-87:B1="E"
1200       if asc(B1)=61 then F=1:A1='f'-87:B1="F"
1210       if F then {Data=X/2
1220         if X and 1 then {
1230              Work(Data)=(Work(Data) and 240)+A1} else {
1240              Work(Data)=(Work(Data) mod 16)+A1*16}
1250              print B1;
1260              X=X+1
1270              if X=16 then F=4
1280              }
1290     endswitch
1300   until F
1310   if F=1 then continue
1320   if F=2 then {Y=Y+1
1330    if Y=16 then {if Pointer>65400 then {Y=15
1340      continue} else {
1350       Out()
1360       Pointer=Pointer-120
1370       if Size<Pointer then Size=Pointer+128
1380       Y=0
1390       break}} else {
1400       locate 6,Y+9
1410       for i=0 to 7
1420         print right$("0"+hex$(Dump(Pointer+i)),2);" ";
1430       next
1440       Pointer=Pointer+8}
1450     break}
1460   if F=3 then {Y=Y-1
1470    if Y=-1 then {if Pointer<118 then {Y=0
1480      continue} else {
1490        Pointer=Pointer-136
1500        Out()
1510        Y=15
1520        break}} else {
1530          locate 6,Y+11
1540          for i=0 to 7
1550           print right$("0"+hex$(Dump(Pointer+i)),2);" ";
1560          next
1570        Pointer=Pointer-8}
1580     break}
1590   if F=4 then {Sum=0
1600     for i=0 to 7
1610       Dump(Pointer+i)=Work(i)
1620       Sum=Sum+Work(i)
1630       Vsum(i)=0
1640     next
1650     locate 33,Y+10
1660     print right$("0"+hex$(Sum),2);
1670     for i=0 to 7
1680       for j=-Y to 15-Y
1690         Vsum(i)=Vsum(i)+Dump(Pointer+j*8+i)
1700       next
1710       locate 6+i*3,27
1720       print right$("0"+hex$(Vsum(i)),2)
1730     next
1740    /* CRC
1750    X=0:Y=Y+1
1760    if Y=16 then {if Pointer>65400 then Y=15 else {
1770       Out()
1780       Pointer=Pointer-120
1790       if Size<Pointer then Size=Pointer+128
1800       Y=0}} else {
1810       Pointer=Pointer+8
1820    break}}
1830   if F=5 then {Pointer=Pointer-Y*8+120
1840    break}
1850   if F=6 then {k=Pointer+X/2
1851          if X mod 2=1 then {
1852            Dump(k)=(Dump(k) and 240)+(Dump(k+1) shr 4)
1853            k=k+1
1854            }
1855       for j=k to Pointer+7-Pointer mod 8
1860         Dump(j)=((Dump(j) and 15)*16+(Dump(j+1) shr 4))
1870       next
1880    Pointer=Pointer-8
1890    Out()
1900    Pointer=Pointer-120
1910    break}
1920   endwhile
1930 if F=5 then Sum=0:break
1940 endwhile
1950 endfunc
1960 func Pr(St;str)
1970   if M then lprint St; else print St;
1980 endfunc
1990 func Prl()
2000   if M then lprint else print
2010 endfunc
2020 func str Crc(P)
2030   int i,j
2040   int A,C,MASK
2050   P=P-120
2060   C=Dump(P)*256+Dump(P+1)
2070   for i=2  to 127
2080     MASK=&H80:D=Dump(P+i)
2090      for j=0 to 7
2100        C=(C shl 1)
2110        if (D and MASK) then C=C+1
2120        if (C and &H10000) then C=C xor &H11021
2130        MASK=MASK shr 1
2140      next
2150   next
2160   return(string$(4-len(hex$(C)),"0")+hex$(C))
2170 endfunc
```

▶いよいよSX-WINDOWのアプリケーションが揃ってきましたね。X68000の新たなる時代に期待しています。　安藤　晶(21)和歌山県

KYOKO IN ANOTHER CG WORLD

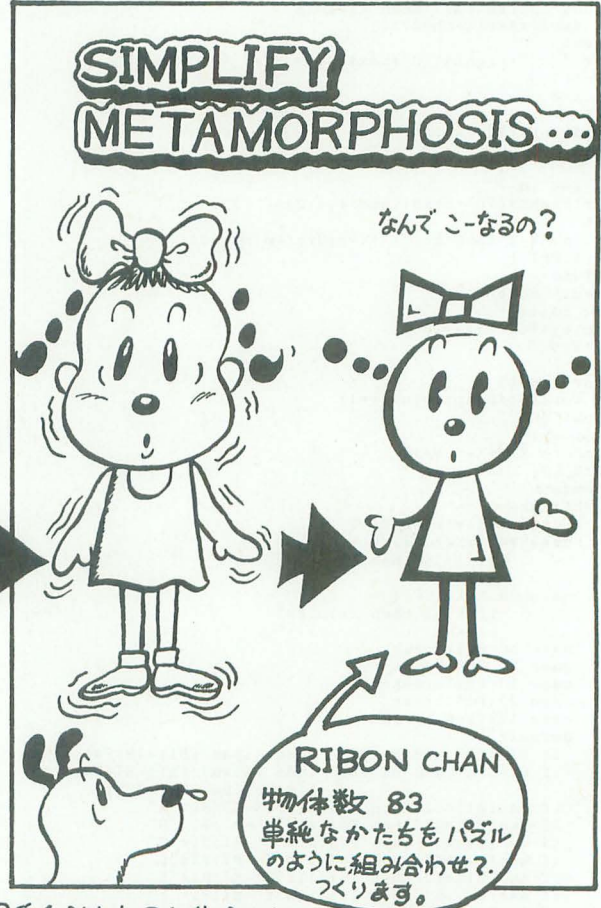
BR::OOOOOOOOOu.

HIKOUSEN
物体数 104
レーザー核融合プラズマジェット
エンジン搭載などと考えると
デザインしやすい…
BODY
だ円体を回転させてつなげる
球体を切ったもの

CG DATA
総物体数 1458
うち星の数 1000 ☆☆☆
光源数 5
1280×1024 PIXEL で出力

RIBON CHAN DATA

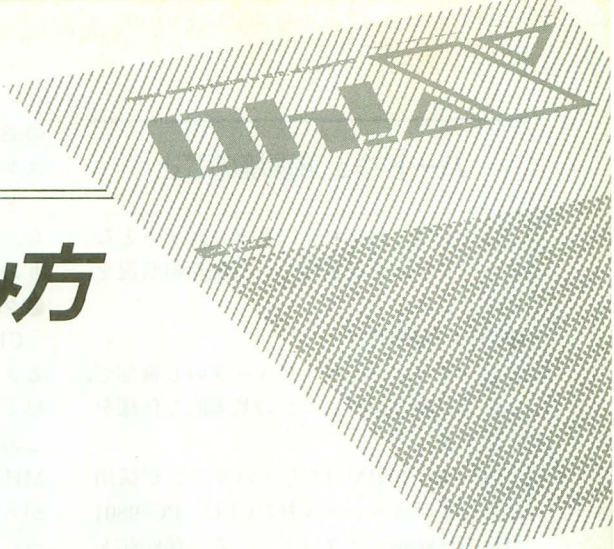
NAME
: RIBON
BODY
: NON CYBOR
DRUG
: NON CONTAIN
DNA.CODE
AGE
: 7

I LOVE X68000…

SIMPLIFY METAMORPHOSIS…

なんでこーなるの?

RIBON CHAN
物体数 83
単純なかたちをパズルのように組み合わせて、つくります。

P.S. その1 芸術祭オープニングの CGA(土田康司氏作)はとってもかっこいい。
その2 M.N.M softwareの「スターモビール」のパッケージCGは X68000で作ったもの(by KYOKO)見てね。

▌特別企画

Oh!Xの正しい読み方

［編集部］

**Oh!Xに正しい読み方など存在しないことはいうまでもない。
とはいえ，心得ていたほうが楽しめることはあるだろう。
知っている人は「その筋事典」の焼き直しかぁ
といった気分でおおらかに読んでいただければ幸いである。**

Oh!Xも創刊されて10年目に入っている。その間，誌名も変わったし，内容的にもいろいろな意味で変化してきたといえるだろう。読者のなかにも，Oh!MZの時代からずーっと読んでくれている人もいれば，ごく最近X68000を買ってから読むようになった人も多いはずだ。ところが，特定の機種を対象とし，毎月続けて読んでくれる読者が多いこともあって，それなりに癖のある誌面になりがちだ。ここでは，Oh!Xがどのような認識に立って誌面作りを行っているかを知ってもらう意味で，通常はあえて触れないような話を集めてみよう。

パソコン本体のこと

まずはハードウェア。Oh!Xだからといって X68000しか出てこないわけではない。頻度の高い他機種も併せて紹介しよう。

●X68000

本誌をお読みの読者諸氏なら知らぬ人はないというくらい有名なパソコンである。発売当初からパーソナルワークステーションと銘打たれていたが，さすがに恥ずかしいのでOh!Xではそういう表現は避けてきた。そう呼べる商品になってもらいたいが，まだ道程は遠い。とはいえ，「理想には」いちばん近いパソコンではないかと編集部内では思われている。自由度はあらゆるパソコンのなかでナンバーワンだろう。もしも負けるとしたらMZ-80Kぐらいか。こだわって使う人にはこたえられないマシンである。

最近のデータでは，本誌読者の9割以上がこのX68000を所有しているらしい。ちなみに，Oh!Xの編集部員でも5人のうち，4人が個人的にX68000を所有している。

X68000には特定の愛称はなく，省略するときも単に「ロクハチ」「エックス」と呼ばれることが多い。一説によるとF社製品のユーザーはなぜかかなりの確率で「ペケロク」と呼ぶという。一部のパソ通愛好者にもいるようだ。Oh!Xの関係者では唯一(で)氏がそう呼んでいる（当人はその事態に気づいている様子がないが)。その他マイナーな呼び方では「ロッパ」「ペケロッパ」などがある。無論，Oh!Xでは正しく「エックスロクマンハッセン」と呼ぶことを推奨しているが，その事実はあまり知れわたってはいない。

●AMIGA

昨年入社した(A)が，AMIGA500を購入したことからブームが起こる。続いて(U)もパワーモンガをやりたいがために購入し，編集部に常備してこれみよがしに海外のソフトを走らせる。512Kバイトのメインメモリと2DDのディスク1枚で，リアルなアニメーションや迫力のPCM音を鳴らすAMIGAのソフトはすごいものがある。そのため，Oh!Xの誌面にはなにかとAMIGAの話題が氾濫している。特に(U)の編集後記（今年の7月号)は，読者を混乱させる悪質なものであった。もちろん誰も反省はしていない。

●Macintosh

ソフトバンクの編集局ではDTPシステムとして活躍しているのがMacintoshだ。Oh!Xでは以前から有田隆也氏が「知能機械概論」などでMacintoshの話題を取り上げているが，最近では荻窪圭氏がMacintosh IIcxを買ったことからよく話題にのぼる。Macintoshは尊敬すべき先輩だが，マック至上主義のユーザーには気をつけよう。彼らは，マック以外はパソコンではないといい，マックはもうパソコンではないともいう。安くなったとはいえ何かしようとするたびにお金がかかるのは変わっていない。

●PC-9801

高価なマシンを買えば，それだけできることが広がるMacintoshと違い，168,000円の16ビット機を買っても70万円の32ビット機を買っても，できることはあまり変わらないのがPC-9801およびその互換機の特長だろう。まあどうでもいいけどね。

Oh!Xオリジナルシステム　COLUMN1

●S-OS "SWORD"

エスオーエスソード。スォード，エスワードは誤り。このシステムがあればTHE SENTINELに掲載されているプログラムが共通に使える。S-OSの意味は不明。現在の編集部には由来を覚えているものはいない。

●MAGIC

マジック。Micro-computer's Advanced Graphic Interpriter Core。

原作はアルシスソフトウェアのTUX吉村氏。Z80のコードを68000に書き直したものが黄金週間PRO-68Kに収録されたバージョン。今月の特集でグラフィックルーチンなどをX68000用に拡張した。拡張しだいで，あと2倍は速くなるかな？

●OPMA/OPMD

オーピーエムエイ/オーピーエムディ。Oh!Xオリジナルの音源ドライバ(？)。従来のOPMDRV.XにMIDI出力とAD PCMの同期機能を拡張するもの。詳しくは今月号の質問箱168ページ参照。

●CARDDRV

カードドライバ。Oh!Xオリジナルのカードゲーム作成用カードデータ管理ドライバ。X-BASICで手軽にカードゲームを作成するためのCARD.FNCをコンパイラ対応に拡張したもの。

●Z's-EX

ジーズイーエックス。X68000でもっとも普及しているグラフィックツール，Z'sSTAFF PRO-68Kの機能を拡張するもの。使用の際はRAM4Mバイトはほしい。

ハードウェア環境のこと

ここでは，パソコンの基本システムとなるハードウェア要素について現在の状況を見ておこう。

●CPU（MPU）

シーピーユー。コンピュータの心臓部で，こいつの能力がパソコンの基本的な仕様を支配する。

PC-9801やIBM-PCなどのマシンが採用しているのはインテル社のCPU。PC-9801の場合は8086でスタートし，その後8086と互換性のあるV30を投入，現在の主流は，16ビットの80286, 32ビットの80386SX/DXとなっている。X68000のCPUはモトローラ社の68000で愛好者は特にMPUと呼んだりする。Macintosh，AMIGA，なども68000マシンだが，主力マシンはいずれも32ビットの68020，68030へと移行している。68030でスタートしたNeXTは現在68040である。

ところで，68系CPUはその生い立ちから，ミニコン下がりの石，86系は電卓上がりの石などと呼ばれることもある。

●クロック

CPUが命令を実行する単位時間を決めるメトロノームのようなもの。クロックが早くなればそれだけ実行速度が速くなる。この春の新製品X68000XVIでは，従来の10MHzから16MHzへとクロック周波数が上がっている。ちなみに現在PC-9801の主力マシンは20MHzの386だが，X68000と同時期に発表になったMZ-2861は8MHzの286であった。なお，無理にクロックを上げることは各自の責任のもとでやってほしい。むやみに他人には勧めないように。

●ハードディスク

かつては贅沢品だったハードディスクもいまやなくてはならない必需品となり，コストも下がっている。すでに日本電気の純正品でも，100Mバイトで128,000円という定価がつけられている。いよいよ10Mあたり1万円が相場となってきた。X68000XVI-HD（81Mバイト）は15万円高で発売3カ月で割高となってしまった。あの内蔵タイプはかなりいい品なのだが。

●メモリ

メインメモリ128KバイトでスタートしたMacintoshも6Mバイトは欲しいという時代になってきた。IBMマシンでもWindows3.0の普及でメモリはいくらあっても困らない。すでに，米国では33MHzの386DXマシンが4MバイトRAMと200MバイトHDDを積んで実売価格は40万円台という状況らしい。

X68000でも，来年はせめて4Mバイト標準機を出す必要があるだろう。

Oh!Xを楽しむための ローカル用語解説

●表紙

ヒョウシ。現在はパーソナルコンピュータによるCGのもの。残念ながら（?）両者ともPC-9801を使用して制作中。須藤氏はパーソナルリンクスとメタエディタ，塚田氏はトランスピュータとサイクロンTachyonまたはC-TRACEを使用。

●ペンギン情報コーナー

ペンギンジョウホウコーナー。ペンギンというのはOh!MZ初期からのキャラクターだが，いつしかイラストは載らなくなった。まあ，それだけのこと。ペンギンのキャラはマシン語カクテルなどで復活している。

●バビンチョ

西川善司氏の合言葉。「まんがはじめて物語」で使われていた呪文から取ったものだ。本人によると「バビンチョ西川」「バビンチョリーナ西川」といった活用形が可能であるとかで注意を要する。

「善バビ」というのは「西川善司のゲームミュージックでバビンチョ」の略。今年の3月号でSOFTWARE INFORMATIONのページに設置されたが，本人の趣味で膨張し，うっとうしい，じゃなかった肝心の新作ソフト情報を圧迫しはじめたため，現在の位置（LIVE in'91）に強制退去，じゃなくて移管された。

●ぽっくん

ポックン。これも西川氏の専門用語。ぽっくん，ぽっくん，とうるさいので現在は使用を控えている。基本的には，なるべくつまらない洒落のあとに「なんちて，ぽっくん」と締めるのが美しい使い方とされている。

●その筋

ソノスジ。かつてOh!MZ最大のキーワードだった。もともとはそんなにいい意味ではなかったはずだが，いつの間にかコンピュータのエキスパートを指す言葉になってしまった。

●桑野式

クワノシキ。グラフィックの階調を落とすためのアルゴリズム。詳しくは8月号参照。

●Oh!X式パレット

オーエックスシキパレット。SX-WINDOWが16色なので，16ビットカラーから変換する際に採用されたGGRBなパレット。RGBのうち，もっとも輝度の高い緑成分を2ビットで処理し，全体のコントラストを抑えめにするとできあがる。できれば標準16色（RGBI）でやりたいが，標準16色用の高速なアルゴリズムが思いつかないので，このまま押し通そうかという動きもある。SXCONV.X, SXIMAGE.Xで採用されている。

●ぜんまいちゃん

ゼンマイチャン。ちょっと古いが，1987年の吉田幸一氏が執筆した「ぜんまい仕掛けのプログラム」（1987年4月号）に登場した女性型アンドロイドの愛称。BASIC入門用としてアンドロイドに命令を与える仮想のインタプリタを想定したもので大反響を呼んだ。3年後のマシン語特集でも再登場している（1990年7月号）。

正確には「全快1号」というハードウェアがあり，その上で走る日本語インタプリタが「ぜんまいちゃんインタプリタV1.0」だったのだが，いつしか全快1号自体がぜんまいちゃんと呼ばれるようになった。かつて，編集部を訪れた女性がゲームソフトの箱を手にとって「これ一太郎で使える？」と言ったそうだ。ぜんまいちゃんのシナリオは，PC-9801が一太郎と呼ばれてしまう世相を反映したものであったのだろうか。

●電子ちゃん

デンコチャン。東京電力の「でんこ」とはまったく関係ない。満開製作所が発行するX68000のためのディスクマガジンでいろんなツールや楽しい絵や音楽や役立つデータがいっぱい入ってしかも電源オンでたちまち起動，あとはマウスでラクラク操作の電脳倶楽部，の広告で本誌読者にはお馴染みだろう。ちなみにあの漫画には，早川くんや石持先輩などは実在の人物とは一切関わりのないキャラクターが登場する。気になるのは8月号から「作：いわいいっぺい」の文字が消えたこと。祝氏が沈黙をやぶって登場するのはいつか。

●祝一平

イワイイッペイ。満開製作所社長にして電脳倶楽部の編集長を務める。もちろん，本誌で活躍していたあの祝一平氏である。経歴については試験に出るX1の著者略歴を参考のこと。解読に成功したらお知らせください。

●満開製作所

マンカイセイサクショ。X68000で年間数10万本以上のソフトを売る会社はここしかない。

●満開一号

マンカイイチゴウ。満開製作所が開発するパーソナルコンピュータの名称。「皿までどーぞ」（1985年3月号）で予約受け付けを開始し，圧倒的な反響を呼んだ。が，当時の編集部内では「ハッタリにしては志が低い」と不評だったという。いま思えばNeXTに酷似しているような気もする。もっともNeXTは立方体，満開一号は流線型なのだが。

●Yet Another Column

イェットアナザーカラム。Oh!X初のディスク付録に収録されたゲーム。わざわざ付録でつけるんだから「絶対に面白いゲーム」ということで選ばれた。PDSのBeyond Columnsを参考に，徹

システムソフトウェアのこと

OS関係についても，とりあえず最低限のものに触れておこう。

●OS

オーエス。IBMの登録商標と聞いたこともあるが，英字2文字が通るはずはない気もする。日本語では朝日新聞が「基本ソフト」などと呼んでいたが，なんのことかさっぱりわからないということで死語になった。主要なデバイス（メモリ，フロッピーディスク，CRT，キーボードなど）の管理，基本的な入出力を一手に引き受けることで，ハードウェア資源を有効に使うことを目的としたシステムのこと。広義にはマシン語モニタからBASICまで，コンピュータを制御する基本プログラムすべてを指す。

●MS-DOS

エムエスドス。マイクロソフト社のOS。X68000とPC-88VAとNAVIを除く16ビット以上の国産パソコンはすべてMS-DOSをベースに動いているといっていい。もともと，インテル社の8088/8086用に作られたもので，286以上のCPUの機能を十分に生かすことができない。次世代のOSとしてOS/2があるが，システムが巨大化したこともあって簡単に移行できるものではない。

●UNIX

ユニックス。いつぞやの日経うんたらの記事中にはX68000の32ビット機の姿としてUNIXが動いてうんぬんと書かれていた。しかし，UNIXが100Mバイトのハードディスクでも動いたのは昔の話。いまやメモリ16Mバイト，ハードディスク300Mバイトくらいが最低線だろうか。それだけ資源を使えばほかになんでもできそうな気がするのだが。一般家庭からJUNETにアクセスす

底的に改良したもの。名前はyacc（Yet Another Compiler Compiler）をもじったもの。

効果音は西川善司氏の担当だが，ブロックを消すときの音は泉氏の好みでなかったため，「チャラララン」という泉氏が作ったものを使用している（西川氏の音は「チントンシャンカシャーン」という感じの音だった）。

現在の推定日本最高記録は56,000点程度と思われる（電卓技を使わない場合）。

●対戦ポピュラス

タイセンポピュラス。昨年の編集室を席巻したゲーム。はたで見ていてこれくらい面白いものはない。西川善司氏は大学でたくさんの友人をなくしたという。これ以降，丹明彦氏の性格が変わったという噂もあるが，スタッフ全体の口が悪くなったのは確か（なぜだろう？）。

それから，SANTIAGOのパスワード教えてくれた方，紹介しそこねてごめん。

●対戦ボンバーマン

タイセンボンバーマン。深夜のマシン室から大声が聞こえる。道行く人は不思議そうな顔をして通りすぎる。山田純二，PCエンジン版5人対戦でOh!PCスタッフを蹴散らす。常勝西川善司「白いボンバーキング」を名乗る（しかも全国2位にランキングされる）。ボンバーマン，ノーコンティニュークリアの金子俊一「赤いボンバーキング」を名乗る。毛内俊行「青い馬鹿」の異名を取る。しかし最強は，ふらっと遊びにきて無敗のまま去っていった横内威至ではないかという噂もある。

●アルシオーネ

アルシオーネ。スバルの4輪駆動車で華門氏の愛車（ただし旧型）。直線で構成されたフォルムの非日本的なスタイル。ただし，異様に床高が高い。「板チョコみたい」「富士山のようだ」と評する人もいる。鮮烈なイタリアンレッドと滅多に見れないスタイルなのでやたら目立つ。ちなみに最近は横断歩道でUターンはしないそうだ。2車線あればブレイキングドリフトでUターン

だって。

●質問電話

シツモンデンワ。日曜日の早朝なんかにかけてくるんじゃない！　誰もいるわけないじゃないか。

●週休2日

シュウキュウフツカ。ちなみにソフトバンク株式会社は完全週休2日制だ。

●付録ディスク

フロクディスク。「ディスクがつく号の2カ月前は薄い」という指摘はけっこう鋭い。わざとやっているわけではないが，人手がいちばん足りなくなるのはその時期だ。適当にあるものを詰め込んでいくのではなく（詰め込んでるけど），そのたびに新しいものを作っているので，前後の号にしわ寄せが出てしまう。やっぱり年1回かな。

●X1

エックスワン。CZ-800C/Dシステムのこと。CZ-801C以降のものは「X1シリーズ」，X1と呼べるのはCZ-800Cのみ。マニアタイプはCZ-801Cが発売されて以降のCZ-800Cを指す。

●68030

ロクハチマルサンマル。どこぞの大学で2台のXVIをバラして30ボードを載せたとかいう噂もあった。沙羅曼蛇が超高速スクロールだそうだ（未確認。問い合わせないこと）。

●インテル係数

インテルケイスウ。インテル社のCPUはインテル社内では，ほかでは再現できない高性能を示すことからつけられた性能換算の目安。贅を尽くしたマシンと門外不出の超高性能コンパイラが秘蔵されているという噂もある。

●資料請求券

シリョウセイキュウケン。広告のすみっこにある3角形のシール。「X68000 Oh!X9係」とか書かれている。どういうわけかこれを編集部に送ってくる人がいるが，編集部ではそういう資料は用意がない。広告にある問い合わせ先に送ってほしい。

●表4

ヒョウ4。表4とはOh!Xの裏面の表紙で，創刊以来ずっとシャープの広告が入っている。広告料金がもっとも高いところだから，もっとも重要な最新機種が陣取るのが普通だが，どういうわけか，やがて消えゆく運命のマイナー機種がここに載るようになる。MZ-1500からX1turboIII，そしてX68000の登場以降は常にX1turboZシリーズの指定席となった。今年になってようやく事態は変わったようだ。現在はXVIが頑張っている。これがSUPERだったらちょっと怖いぞ。

●原稿料

ゲンコウリョウ。いきなり電話をかけてきて，原稿料は1字いくらですか？　とくる人がいるのには結構驚かされる。そういう人にOh!Xの原稿を書いてもらうわけにはいかない。そりゃ，高い人も安い人もいるし，内容によりけりである。まあ，1ページ書けばゲームソフトの1本ぐらいは余裕で買えるでしょうけど。昨年の山田純二くんのように原稿料でX68000を買って，ますます原稿を書く（プログラムを作る）といった心意気がほしいな。

●年末/ゴールデンウイーク/お盆進行

ネンマツ/ゴールデンウイーク/オボンシンコウ。世間には迷惑な習慣があって，人々はいっせいに休暇をとるようになっている。もちろん，印刷，流通業界関係者も同じである。ところが，Oh!Xは毎月必ず発売になる。そこで生じるのが〜進行という前倒しのスケジュールである。しかもその傾向は年々厳しくなっているのが現状だ。たとえば今月号の場合，本の完成はいつもの月より4日も早いのに書店での発売日は19日の月曜日になってしまった。予告を見ずに18日に買いにいかれたかたごめんなさい。

ところで，「荻窪圭にゴールデンウイークはない」と言っていた人がいたが，ゴールデンウイークとは毎日会社勤めをしている人が使う言葉だと思う。うんうん。

るというのは，ほんの少しの例しか聞いたことがないし……。

●OS-9

オーエスナイン。マルチタスクがウリのOS。古くはMZ-80K用OS-9の入門を初め，Oh!MZ/X誌上ではCP/M-80に比べても破格に優遇されたOSだといえる。

●CP/M,P-CP/M

シーピーエム，ピー（パーソナル）シーピーエム。8080,Z80で世界標準となったOS。一時は世界を制覇した。

●Human68k

X68000のオリジナルOS。発表当時，どうしてMS-DOSを採用しなかったのかといった質問をあちこちで耳にした。そんなことをいわれても，MS-DOSは86系CPUのOSだ。68000上で動くMS-DOSはマイクロソフトにはなかったのだからしょうがない。そこで作られたのが，Human68kというわけだ。MS-DOSとファイルフォーマットを合わせ，コマンド操作もあらかた上位互換である。また，ファンクションコールも同様の配列をとるなどの配慮がされている。

製品紹介について

雑誌の記事でもっとも気を使うのは製品紹介だ。もちろん情報を正確に伝えなくては読者にもメーカーにも迷惑をかけることになるのは当たり前だが，もっと問題なのは，数値のような客観的な情報だけでなく，それをどう評価するかといったことである。また，ゲームソフトのようにもともと数値などでは表せないものについてもなにかと難しい。

●GAME REVIEW

ゲームレビュー。Oh!Xのライターは不器用なので，悪いところもハッキリ書いてしまい結構わかりやすいと評判。また，きつく書かれていてもいい場所で紹介されていれば全体的な評価は高いと見てよい。ゲーム専門誌では，表現が技巧的で善し悪しは判断しづらいが，買っちゃいけないものは行間にちゃんとそう書いてある（一部を除く）。

表1 Oh!XにおけるXfamily新製品記事

年度	月号	機種名	ページ数
1987	12	X1twin X1turboZII X68000(ブラック)	2 (特集) 2 1
1988	4	X68000ACE-HD	1 (カラー)
1988	5	X68000ACE	1 (カラー)
1988	12	X1turboZIII	1 (ペンギン)
1989	4	X68000PRO/EXPERT	5 (カラー)
1990	5	X68000PROII/EXPERTII/SUPER-HD	8 (カラー＋2色)
1991	1	X68000SUPER	1/3 (ペンギン)
1991	5	X68000XVI	16 (特集)

●新機種

X68000を買ったら1週間後に新機種が出たと嘆く人がいる。確かに，HDなしのSUPERはまずかった。XVIを予想せずに買った人はショックだったと思う。そういう人は表1をよく見ておいてほしい。毎月注意深くOh!Xを読んでいればそのような事態は避けられるのではないかと思う。怪しいと思ったら，新機種を大きく取り上げないこともあるのだ。もっとも，X68000の場合は新機種が出ても直前に買った旧機種で後悔する必要はほとんどなかった。XVIを除けばすべての機種がメモリの増設やシステムのバージョンアップで同等の機能を持つようになるからだ。新製品が出るのを待って安くなった旧機種を狙う人も多い。

読者参加

●STUDIO X

古くは「読者から編集室へ」というコーナーで，のちに「STUDIO MZ」となり，現在の「STUDIO X」に至る。その際につけた「ON AIR」という飾りのパターンが強すぎて，いまだに「STUDIO ON AIR」というページだと誤解している人があとを断たない。

STUDIO Xやハミダシに載るためにはアンケートハガキの「編集部へのメッセージ」の欄に記入すればよい。いちばんの困ったちゃんは，「僕は毎月ハガキを出しているのにまだ一度も載ったことがありません。お願いです，今度こそぜっっったいに載せてください！」と書かれたもの。たいてい肝心なメッセージがどこにも書かれていない。どうすればよいのでしょうね。

というわけで，たいして役にも立たなかったかもしれないが，今後も疑問に思うことがあれば，アンケートハガキでお寄せください。お待ちしております。

Oh!X「数字の秘密」

●公称発行部数

Oh!Xの公称発行部数は一応10万部だ。しかし，本当はそれよりも結構少ない。ところで雑誌の公称部数というと実際の2倍くらいまでは許されることが多いが，X68000ユーザーの過半数が読んでいるOh!Xでは2倍もするとユーザーよりも読者のほうが多くなってしまい，信用問題になる。そこで相手によってはむしろ9万部とか8万部とか7万5千部とか少なくいうようになるわけだ。本当はそれよりももうちょっとだけ少ない。

●特別定価

特別定価というのは，結構便利な言葉で，特別に高い場合と，特別に安い場合とがある。高くなってごめんねという意味と，この値段に抑えているんだよという意味を同時に持っている場合もある。また，値上げの前触れとしても利用されることが多い。たとえば，最近のOh!Xの場合，

4月号まで　定価560円
5月号　特別定価780円　ディスク付き
6月号　特別定価600円　9周年増ページ
7月号　特別定価600円　ポケットブック付き
8月号から　定価600円

となっている。こうしてうやむやのうちに定価を上げてしまうのが一般的な手法である。もっともOh!Xの読者は固定層だからそんな手口は通用しない。男らしくどーんと値上げしたほうがいいという意見が多いのには驚かされた。次の値上げはどーんと1,000円ぐらいとか？

X68000用
One

Endoh Ryuichi
遠藤 隆一

X1用
WHITE MANE

Abe Toshimitsu
阿部 俊光

今月は，なぜかメジャーなアーティストの組み合わせとなってしまいました。ただし，2曲ともアルバムからの選曲なので，知らない方もいるかもしれませんけど。とはいえ，曲自体も作品の出来も上々なので，夏休みをしめくくる大仕事としてぜひがんばって打ち込んでみてください。

たったひとつの……

今月のX68000用は，PRINCESS PRINCESSの「One」をお届けしましょう。OPMD用です。

この曲はアルバム「PRINCESS PRINCESS」のトリをつとめる曲ですね。作曲者の奥居香さんは，ヴォーカルをメインに小物を担当しているお姉さんです。その昔，PRI²が赤坂小町と名乗っていたころからのメンバーですね。

さてと，作品の解説とまいりましょう。いきなりですが，この作品では音色の煮詰めにやや甘さが残っているようです。たとえば，「やっと最後の恋にめぐり逢えた」という歌詞がついている歌にしては，ヴォーカルの音が硬いと思いませんか。原曲では，確かに奥居さんの声量でかなり強い声に聞こえるのですが，もっと優しい音にしたほうがよいでしょう。オープニングのピアノにしても，ちょっと狙いすぎです。もっと素直なピアノに徹したほうがきれいに仕上がると思いますよ。

それから，サンプリングのCONFIGファイルに若干の変更があります。普通にOPMDで鳴らすとオーケストラヒットが鳴りますので，

15=sd5v13.pcm
23=bd1v15.pcm
43=crsh2v8.pcm

に変更してください。変更したくない場合は一度入力したあとで，エディタなどで置換をしてみてください。

それと，たいしたことではないのですが，テンポはひとつのトラックに書いておけば，ほかのトラックもそれにあわせたテンポになります。この作品では5000行あたりにテンポの山がありましたので，その部分だけは削除してあります。このほうが理論的にもテンポずれは減るはずです。

PRINCESS PRINCESS

作者の遠藤君は16歳で，初投稿ということです。これからも月1，2で作品を送っていただけるということで，非常に楽しみにしております。もっといろいろ研究して，素晴らしい作品を作ってくださいね。

ちょっと長めのリストですが，がんばって入力しましょうね。

X1用が載っているぅ

嬉しいことにX1用の作品を掲載します。最近は諸般の事情もさることながら，X1の投稿作品自体少なくなってきているので，あまり掲載されていなかったですからね。

さて，曲はT-SQUAREの「WHITE MANE」です。ホワイト・マネではありませんよ。ホワイト・メインと読むようです（が，私には読めません)。MusicBASIC用です。T-SQUAREはこのLIVE inでも有名ですので，とくに紹介するまでもないでしょう。5人組のフュージョンバンドですよね。F1のイメージソング「TRUTH」はあまりにもメジャーです。この曲はアルバム「ナチュラル」からの選曲になりますね。世の中にはU.S.バージョンというのも存在するようですが，こちらはノーマルの「WHITE MANE」のようです。

ナチュラル

作者の阿部君は3月号に掲載されたLINDBERGの「LITTLE WING」以来の2回目の登場になりますね。

さすがにT-SQUAREの曲を再現するのはちょっと大変なのでしょう。もともとフュージョンは，テクニック命ともいえるジャンルですし，音数も足りないのは目に見えてます。それでも投稿は多いんですよね。この作品では足りない音数をごまかすために，ちょっとしたテクニックが盛り込まれています。これは，コードを省略するときには，ほかのパートで鳴っている音を抜いてしまうという方法です。具体的には，ベースと同じ音を和音から抜きます。たとえば，Cのコードは，C，E，Gの3音で成り立っていますが，ベースがCのときはE，Gの2音を鳴らせばよいということです。ベースの音の倍音や3倍音とコードが和音になるということなのですが，わかっていただけますか。この方法は，ピアノやオルガンなどで同じ音色の場合には非常に有効です。知らなかった人は曲作りの参考にしてみてください。

作品の出来は，苦労のしがいもあってなかなか聴きごたえのあるものに仕上がっています。曲が長いこともあって，ちょっと中ダレを感じますが，ここまでできていればOKですよね。 (S.K.)

日本音楽著作権協会(出)許諾第9170916-101号

リスト1 One

```
  10 /*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*
  20 /*                                                 /*
  30 /*                    One                          /*
  40 /*                                                 /*
  50 /*                                                 /*
  60 /*                                                 /*
  70 /*                          Princess Princess      /*
  80 /*                                                 /*
  90 /*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*
 100 m_init():m_tempo(130)
 110 for i=1 to 8:m_alloc(i,5000):m_assign(i,i):next
 120 str a(50)[256],b(50)[256],c(50)[256],d(50)[256],e(50)[256]
,f(50)[256],g(50)[256],d4[20],d8[20],d6[20],d24[20],s4[20],s8[20
],s6[20],h4[20],h8[20],h6[20],l4[20],l8[20],l6[20],cr[20],cy[9]
 130 str h(50)[256],k[256],w4[10],w8[10],w6[10],l9[20]
 140 key 19,"m_tempo(200)"+chr$(13):key 20,"m_tempo(130)"+chr$(
13)
 150 /*
 160 /*               VOICE  SET
 170 dim char v(4,10)={                          /*Strings
 180 53,15,0,0,0,0,0,0,0,3,0,
 190 31,0,0,0,0,17,0,2,7,0,0,
 200 13,0,0,6,0, 7,0,6,2,0,0,
 210 12,0,0,6,0, 0,0,6,2,0,0,
 220 12,0,0,5,0, 0,0,2,6,0,0}
 230 m_vset(11,v)
 240 v={                                         /*BASS
 250 58,15,0,0, 0,0,0,0,0,3,0,
 260 31,12,1,0,15,25,0,0,7,0,0,
 270 31,0, 0,9, 0,50,0,4,3,0,0,
 280 20,10,1,9,15,27,0,0,3,0,0,
 290 20,10,0,7, 0, 0,0,0,7,0,0}
 300 m_vset(2,v)
 310 v={
 320 60,15,0,0,0,0,0,0,0,3,0,
 330 31,8,0,0,2,37,0,8,7,0,0,
 340 15,0,0,6,0, 0,0,8,7,0,0,
 350 31,8,0,0,2,37,0,8,3,0,0,
 360 15,0,0,6,0,0, 0,8,3,0,0}
 370 m_vset(4,v)
 380 v={                                         /*PIANO
 390 60,15,0,0,0, 0,0,0,0,3,0,
 400 31, 0,0,0,0,27,0,2,7,0,0,
 410 24,10,4,4,8, 0,0,1,7,0,0,
 420 31, 0,0,0,0,27,0,3,3,0,0,
 430 31,10,4,4,8, 0,0,1,3,0,0}
 440 m_vset(5,v)
 450 v={                                         /*HH, OP C+
 460 51,15, 0,0,0, 0,0,0,0,3,0,
 470 26, 0, 0,0,0,17,0,1,7,1,0,
 480 26, 4, 0,0,2, 8,0,7,0,0,0,
 490 31,18, 1,0,4,21,0,1,0,2,0,
 500 31,11,12,6,4, 5,0,1,3,0,0}
 510 m_vset(8,v)
 520 v={                                         /*GUITAR
 530 60,15,0,0,0,0,0,0,0,3,0,
 540 31,12,0,0,8,20,0,2,7,0,0,
 550 31, 8,5,4,5, 0,0,2,7,0,0,
 560 22, 0,0,0,0,20,0,3,3,0,0,
 570 31,10,5,4,5, 0,0,2,3,0,0}
 580 m_vset(9,v)
 590 v={                                         /*Horn
 600 44,15,0,0,0,0,0,0,0,3,0,
 610 31,0,0,0,0,22,0,1,7,0,0,
 620 16,0,0,6,0, 0,0,1,7,0,0,
 630 31,0,0,0,0,17,0,1,3,0,0,
 640 13,0,0,6,0, 0,0,1,3,0,0}
 650 m_vset(10,v)
 660 v={                                         /*Low  str
 670 61,15,0,0,0,0,0,0,0,3,0,
 680 31,5,0,0,2,35,0,1,7,0,0,
 690 10,0,0,6,0, 0,0,2,3,0,0,
 700 11,0,4,6,0, 0,0,4,0,0,0,
 710 10,0,0,6,0, 0,0,2,3,0,0}
 720 m_vset(12,v)
 730 v={                                         /*Vocal
 740 44,15,0,0,0,0,0,0,0,3,0,
 750 31,0,0,0,0,17,0,4,7,0,0,
 760 22,0,6,6,0, 3,0,8,7,0,0,
 770 31,0,0,0,0,16,0,4,3,0,0,
 780 22,0,6,6,0, 3,0,8,3,0,0}
 790 m_vset(1,v)
 800 v={                                         /*Flute mh
 810 59,15,0,0,0, 0,0,0,0,3,0,
 820 31,20,3,0,2, 0,0,2,3,0,0,
 830 31,10,0,0,1,77,0,2,7,0,0,
 840 15,10,0,0,0,42,0,2,0,0,0,
 850 13,10,0,5,0, 0,0,2,0,0,0}
 860 m_vset(13,v)
 870 v={                                         /*Backing
 880 52,15,0,0,0,0,0,0,0,3,0,
 890 11,0,0,0,0,37,0,4,7,0,0,
 900  8,0,0,3,0,0, 0,8,7,0,0,
 910 11,0,0,0,0,37,0,4,3,0,0,
 920  8,0,0,3,0,0, 0,8,3,0,0}
 930 m_vset(14,v)
 940 v={                                         /*Alpegio
 950 52,15,0,0,0,0,0,0,0,3,0,
 960 31,8,0,0,3,22,0,8,7,0,0,
 970 22,0,0,6,0, 0,0,8,7,0,0,
 980 31,5,0,0,2,22,0,8,3,0,0,
 990 15,0,0,6,0, 0,0,8,3,0,0}
1000 m_vset(15,v)
1010 v={                                         /*Tom
1020 60,15,0,0,0,0,0,0,0,3,0,
1030 31, 0,0, 0,1,12,0,1,7,0,0,
1040 31,15,15,6,5,0, 0,2,7,0,0,
1050 31,22,13,0,8,7, 0,3,3,0,0,
1060 31,15,15,6,5,0, 0,1,3,0,0}
1070 m_vset(16,v)
1080 v={                                         /*Snare
1090 60,15, 0,0, 0,0,0,0,0,3,0,
1100 31, 9, 8,0,10,7,0,6,7,0,0,
1110 31,15,14,6, 1,0,0,1,7,0,0,
1120 31,20,10,0,10,7,0,5,3,0,0,
1130 31, 0,14,6, 0,0,0,1,3,0,0}
1140 m_vset(17,v)
1150 v={                                         /*BD
1160 59,15, 0,0,0, 0,0,0,0,3,0,
1170 31,20,20,0,8,12,0,1,0,0,0,
1180 31,21,20,0,8,17,0,0,0,0,0,
1190 31,20,20,0,8,10,0,1,0,0,0,
1200 31,11,10,8,15,0,0,1,0,0,0}
1210 m_vset(18,v)
1220 v={
1230 60,15,0,0,0,0,0,0,0,3,0,
1240 31,0,0,0,0,27,0,6,7,0,0,
1250 11,0,1,7,0, 0,0,8,7,0,0,
1260 31,0,0,0,0,27,0,6,3,0,0,
1270 11,0,2,7,0, 0,0,8,3,0,0}
1280 m_vset(19,v)
1290 v={
1300 44,15,0,0,0,0,0, 0,0,3,0,
1310 31, 5,6,0,5,0,0,15,7,0,0,
1320 31,10,4,5,4,0,0, 3,7,0,0,
1330 31,20,6,0,3,0,0, 1,3,3,0,
1340 31,25,6,5,7,0,2, 3,3,0,0}
1350 m_vset(20,v)
1360 v={                                         /*M Guitar
1370 59,15,0,0,0, 0,0,0,0,3,0,
1380 31, 0,0,0,4,32,0,7,7,0,0,
1390 31,20,0,0,3,17,0,6,7,0,0,
1400 22,18,0,0,5,25,0,4,3,0,0,
1410 31,10,5,3,5, 2,0,1,3,0,0}
1420 m_vset(21,v)
1430 v={                                         /*Crush
1440 44,15,0,0,0,0,0,0,0,3,0,
1450 31, 5,6,0,5,0,0,15,7,0,0,
1460 31,10,4,4,4,5,0, 3,7,0,0,
1470 31,20,6,3,3,0,0, 1,3,3,0,
1480 31,25,6,4,7,5,2, 3,3,0,0}
1490 m_vset(22,v)
1500 /*                  DRUM  SET
1510 d8="y2,23r8": s8="y2,15r8": d4="y2,23r4": s4="y2,15r4"
1520 /*d8="@18@v127o1c8": s4="@17@v127o1e4": d4="@18c4": s8="@v
126o1@17e8"
1530 k="@16r2r8":s6="y2,15r16":d6="y2,23r16":d24="y2,23r24":h4=
"@16f4":h8="@16f8":h6="@v126o1@16f16"
1540 /*k="@16r2r8":s6="@17e16":d6="@18c16":d24="@18c24":h4="@16
f4":h8="@16f8":h6="@16f16"
1550 l4="@16d4":l8="@16d8":l6="@126o1@16d16":l9="@16o1@v127d8."
:w4="r4":w8="r8":w6="r16":cr="@8@v126o5p1c4o1p3":cy="y2,43r4"
1560 /*l4="@16d4":l8="@16d8":l6="@16d16":l9="@16o1@v127d8.":w4=
"r4":w8="@v126o1r8":w6="r16":cr="@8o5c4o1":cy="@8o5>e4o1"
1570 /*
1580 /* 1660,1680,1700行は FM Drumsです。1650,1670,1690 にか
えるとADPCMになります。
1590 /*
1600 /*                         A
1610 /*
1620 m_trk(1,k+d8+s4+cr+s8+d8+d4+s4)
1630 a(1)="@2v13o3L8q7 >r2r4b&b-16&a+16 ffffffff
1640 b(1)="@5v13o4L8y50,20p1 r1 <c>fa<c&c2
1650 c(1)="@5v10o4L8 r16  p3 r1 <c>fa<c&c2
1660 d(1)="@5v13o4L8y52,00p2 r1 <c>fa<c&c2
1670 e(1)="@22v15o5L8        p3 r1 g@8v13p2o4ggggggg
1680 f(1)="@21v13o2L8        p3 r1 <f>ff<f>ff<f>f
1690 g(1)="r1 r1
1700 m_trk(2,a(1))
1710 m_trk(3,b(1))
1720 m_trk(4,c(1))
1730 m_trk(5,d(1))
1740 m_trk(6,e(1))
1750 m_trk(7,f(1))
1760 m_trk(8,g(1))
1770 /*
1780 /*                         B
1790 /*
1800 m_trk(1,d4+s4+d8+d8+s4+d4+s8+d8+d4+s4+d4+s4+d8+d24+d24+d24
+s4+d4+s8+d8+d4+s4):m_trk(1,d4+s4+d8+d8+s4+d4+s8+d8+d4+s4+d4+s8+
d8)
1810 m_trk(1,l9+s6+s8+l8)
1820 a(2)="ffffffff ffffffff fffffff<f ffffffff ffffffff ffffff
ff fffffff&f-16&e+16
1830 b(2)="|:2d>fb-<d&d2 e>g<ce&e2 d>fb-<d&d2|1c>fa<c&c2:||2
1840 c(2)="|:2d>fb-<d&d2 e>g<ce&e2 d>fb-<d&d2|1c>fa<c&c2:||2
1850 d(2)="|:2d>fb-<d&d2 e>g<ce&e2 d>fb-<d&d2|1c>fa<c&c2:||2
1860 e(2)="|:v15gv13q6ggggggg:|v15q8gv13q4ggggq8v15g16g16v13p1@
22o5e4 e@8o4v13p2ggggggg |:v15gv13q6ggggggg:|ggggr2
1870 f(2)="@21v13L8o2|:7<f>ff<f>ff<f>f:|
1880 g(2)="r1 r1 r1 r1 r1 r1 r1
1890 m_trk(2,a(2))
1900 m_trk(3,b(2))
1910 m_trk(4,c(2))
1920 m_trk(5,d(2))
1930 m_trk(6,e(2))
1940 m_trk(7,f(2))
1950 m_trk(8,g(2))
1960 /*                         C      .
1970 /*
1980 m_trk(1,cr+s8+d8+d4+s4+d4+s8+d8+d4+s4+d4+s4+d4+s4+d4+s4+d8
+d8+s4)
1990 a(3)="@2v13o3L8 >|:3ffffffff:|<ffffffff >
2000 b(3)="@1v15o3L8y50,00p3 r4rcf4cd& d1 r4.ggab-a& a4. gg4. f
```


▶いま,受験生の私は「いかにして親にばれずにパソコンを触るか」という課題に取り組んでいます。
児島 広(18)奈良県

```
 2010 c(3)="@1v13o3L8y51,30p3 r4  r   cf4cd& d1 y51,40v15r4.ggab
-a& a4. gg4. f
 2020 d(3)="@5o4v11L8p3y52,24 v13f1 p1v11f>fb-<ddf p3v13a4 g2&g
p1v11ceg  p3v13a p1v11cfap3v13gp1v11cp3v13<c4>
 2030 e(3)="@22o5v15L8p3q8e@8o4v13p2v11q6ggggggg |:3v13q8gv13q6g
gggggg:|
 2040 f(3)="@21v15o2L8p3 |:3fffffff:| ffffeeee
 2050 g(3)="@5o4v1118p3v13 c1 d>p2v11fb-<ddfp3v13f4 e2&ep2v11ceg
 p3v13fp2v11cfap3v13ep2v11cp3v13b-4
 2060 m_trk(2,a(3))/*BASS
 2070 m_trk(3,b(3))/*VOCAL
 2080 m_trk(4,c(3))/*VOCAL
 2090 m_trk(5,d(3))/*PIANO
 2100 m_trk(6,e(3))/*H.HAT
 2110 m_trk(7,f(3))/*GUITAR
 2120 m_trk(8,g(3))/*PIANO
 2130 /*
 2140 /*                        4
 2150 /*
 2160 m_trk(1,d4+s4+d8+d8+s4+d4+s4+d4+s4+d4+s8+d8+d4+s4+d4+s4+w8
+18+s8+18)
 2170 a(4)="<dddddddc >bbbbbbbf ggggggg <cccccc<c>c >
 2180 b(4)="f4      fe4f4a& ag4d4e4f& f2rfff fedc2&c8
 2190 c(4)="f4y51,30v13fe4f4a& ag4d4e4f& f2rfff fedc2&c8
 2200 d(4)="p3a2&ap1v11ef<c> p3v13b4.gg2 f2&fp1v11>gb-<d p3v13b-
4b-b-&b-2
 2210 e(4)="|:4v14q8gv14q6gggggg:|
 2220 f(4)="L8dddddddd >bbbbbbb ggggggg <cccccccc
 2230 g(4)="p3f2&fp2v11ef<c> p3v13g4.dd2 d2&dp2v11>gb-<d p3v13f4
fe&e2
 2240 m_trk(2,a(4))/*BASS
 2250 m_trk(3,b(4))/*VOCAL
 2260 m_trk(4,c(4))/*VOCAL
 2270 m_trk(5,d(4))/*PIANO
 2280 m_trk(6,e(4))/*HHAT
 2290 m_trk(7,f(4))/*GUITAR
 2300 m_trk(8,g(4))/*PIANO
 2310 /*
 2320 /*                        5
 2330 /*
 2340 m_trk(1,cy+s8+d8+d4+s4+d4+s4+d8+d8+s4+d4+s4+d4+s4+d4+s8+d8
+d4+s4+d4+s4+d4+s4):
      m_trk(1,d4+s8+d8+d4+s4+d4+s8+d8+d4+s4+d8+s8+s8+s6+s6+s6+s6
+h6+h6+18+s8)
 2350 a(5)="|:3fffffff:| <ffffeeee dddddddc >bbbbbbbf gggggab-<
d ccccc>a<dc
 2360 b(5)="r4.cf4cd& d1 r4 gggab-a& a4.gg4.f f4fe4f4a& ag4d4e4f
& f2rfff fefg2f
 2370 c(5)="p3r4.cf4cd& d1 r4 y51,40v15gggab-a& a4.gg4.f f4 y51,
30v13fe4f4a& ag4d4e4f& f2rfff fefg2f
 2380 d(5)="f1 fp1v11>ab-<ddfp3v13a4 g2&gp1v11ceg p3v13ap1v11cfa
p3v13gp1v11cp3v13<c4> a2&ap1v11ef<c> p3v13b4.g2&g8 f2fp1v11>gb-<
d p3v13<c4.>b-&b-2
 2390 e(5)="@22o5v15p3e@8o4v13p2q7ggggggg |:6gggggggg:|gggr8r4r8
@22o5v15p3e@8o4p2
 2400 f(5)="|:3fffffff:|ffffeeee ddddddd >bbbbbbb ggggggg <c
ccccccc
 2410 g(5)="c1 dp2v11>ab-<ddfp3v13f4 e2&ep2v11ceg p3v13fp2v11cfa
p3v13ep2v11cp3v13b-4  f2&fp2v11ef<c> p3v13g4.d2&d8 d2dp2v11>gb-<
d p3v13f4.e&e2
 2420 m_trk(2,a(5))
 2430 m_trk(3,b(5))
 2440 m_trk(4,c(5))
 2450 m_trk(5,d(5))
 2460 m_trk(6,e(5))
 2470 m_trk(7,f(5))
 2480 m_trk(8,g(5))
 2490 /*
 2500 /*                        6
 2510 /*
 2520 m_trk(1,d4+s8+d8+w4+s4+d4+s8+d8+w4+s8+d8+d4+s8+d8+w4++s8+d
8+d4+s8+d8+w8+d6+d6+s8+d8)
 2530 a(6)="o3>b-4.<b-4.>b-4 a4.<a4>ab-a <e-4.<e-4.>e-4 d4.<d4.>
>b-a
 2540 b(6)="v15p3  e4ff4g4f& fc4a2a a4gg4f4a& aggf4.df
 2550 c(6)="v14p3  e4ff4g4f& fc4a2a a4gg4f4a& aggf4.df
 2560 d(6)="@5v13o4L8a4.a2&a a4.a2&a <c4.c4c4. c4.c4c4.>
 2570 e(6)="L4@22o5v15p3e@8o4v14q6p2ggg |:q8v15gp2v14q6ggg:|q8gq
6ggg
 2580 f(6)="@9v13o3L8p1 fb-<cf4v12c>b-4 v13fa<cf4v12c>a4 v13e-a<
cf4v12c>a4 v13da<cf4v12c>a4
 2590 g(6)="@5v13o4L8d4.d2&d c4.c&c2 f4.f4f4. d4.d4d4.
 2600 m_trk(2,a(6))
 2610 m_trk(3,b(6))
 2620 m_trk(4,c(6))
 2630 m_trk(5,d(6))
 2640 m_trk(6,e(6))
 2650 m_trk(7,f(6))
 2660 m_trk(8,g(6))
 2670 /*
 2680 /*                  7
 2690 /*
 2700 m_trk(1,d4+s8+d8+w8+d8+s4+d4+s8+d8+w8+d8+s4+d4+s8+d8+w8+d8
+s4):m_trk(1,d8+s8+s8+d8+s8+s8+s8+d8)
 2710 a(7)="L4g.<g>g8a8g8 g-.<g-.>g- f.<f.>f <c8cgc8<c8>c8
 2720 b(7)="p3 a+16&b-8.b-b-4b-4a& ab-4<c4>b-4b- b-4ag4d4f& fefg
2&g
 2730 c(7)="p3 a+16&b-8.b-b-4b-4a& ab-4<c4>b-4b- b-4ag4d4f& fefg
2&g
 2740 d(7)="L8f2..v13a& ab-4<c&c2> f1 v13 b-4.b-&b-2
 2750 e(7)="|:4p2v15q8gv14q7ggg:|
 2760 f(7)="p1v13f<c>b-<f4v12c>b-4 v13f+a<df+4v12d>a4 v13f<c>b-<
f4v12c>b-4v13f4.g&g2
 2770 g(7)="d2..v13f+8& f+8f+4a8&a2 d1 v13 f4.e8&e2
 2780 m_trk(2,a(7))
 2790 m_trk(3,b(7))
 2800 m_trk(4,c(7))
 2810 m_trk(5,d(7))
 2820 m_trk(6,e(7))
 2830 m_trk(7,f(7))
 2840 m_trk(8,g(7))
 2850 /*
 2860 /*           8
 2870 /*
 2880 m_trk(1,d4+s8+d8+w8+d8+s8+d8+d4+s4+d8+d8+s8+d8+d4+s8+s6+s6
+s8):m_trk(1,h8+18+18+cy+w8+w8+w8+w8+w8+w8)
 2890 a(8)="o3>L8b-b-b-b-b-b-b-b- aaaaaaa gggggab-b <c2.&c16&c+
16&d-16&c+16
 2900 b(8)="p3L4ef8f2&f8 ef8v14fgv15<c8& c1 >v15b-ade8f8
 2910 c(8)="p3L4ef8f2&f8 ef8v14fgv15<c8& c1 >v14b-ade8f8
 2920 d(8)="@15o2v12p2L8 b-<cfc4cef >a<cfc4cef >b-<cfc4cefb-1
 2930 e(8)="L8|:p2gggggggg:|L4q8gq4gr2 @22o5q8p3v15e2.e4@8o4
 2940 f(8)="fb-<cf4v12c>b-<c v13>fa<cf4v12c>a<c v13>fb-<cf4v12c>
b-<c v13c1
 2950 g(8)="@15o2v10p2L8 r8  b-<cfc4cef >a<cfc4cef >b-<cfc4ce
v11g1
 2960 m_trk(2,a(8))
 2970 m_trk(3,b(8))
 2980 m_trk(4,c(8))
 2990 m_trk(5,d(8))
 3000 m_trk(6,e(8))
 3010 m_trk(7,f(8))
 3020 m_trk(8,g(8))
 3030 /*
 3040 /*                 9
 3050 /*
 3060 m_trk(1,d4+s4+d4+s4+d4+s4+d8+d8+s4+d4+s4+d8+d8+s4+d4+s4+d8
+d8+s4):m_trk(1,d4+s4+d8+d8+s4+d4+s4+d8+d8+s4)
 3070 a(9)="o3>|:3fffffff:|<ffffeeee dddddddc >bbbbbbb<
 3080 b(9)="r4.L8cf4cd& d1 r4.ggab-a& a4.gg4.f f4fe4f4a& ag4gd4e
f&
 3090 c(9)="p3r4.L8cf4cd& d1 r4.y51,40v15ggab-a& a4.gg4.f f4y51,
30v13fe4f4a& ag4gd4ef&
 3100 d(9)="@5o4v11L8p1 <c>L16{a<c>a}8L8fc>a<cda b-dfb-<d>b-fd e
ceg<c>gec acfaeg<c>e f<ed>afedc >b<dap3g&g2
 3110 e(9)="L8v15p2gv14ggggggg |:3gggggggg:|@22o5v15e@8o4v14p2gg
ggggg egggggggg
 3120 f(9)="@21o2v13p3|:3fffffff:| ffffeeee ddddddd >bbbbbbb
 3130 g(9)="@5o4v11L8p2 <c>L16{a<c>a}8L8fc>a<cda b-dfb-<d>b-fd e
ceg<c>gec acfaeg<c>e f<ed>afedc >b<dap3d&d2
 3140 m_trk(2,a(9))
 3150 m_trk(3,b(9))
 3160 m_trk(4,c(9))
 3170 m_trk(5,d(9))
 3180 m_trk(6,e(9))
 3190 m_trk(7,f(9))
 3200 m_trk(8,g(9))
 3210 /*
 3220 /*                    10 END
 3230 /*
 3240 m_trk(1,d4+s4+d8+d8+s4+d8+s8+s8+s6+s6+s6+s6+s6+s6+s8+d8):m
_trk(1,d4+s8+d8+d4+s4)
 3250 a(10)="@2v13L8o3ggd4>gg<d4 cccccc<c>c >ffffffff
 3260 f(10)="@1v15o3L8y54,30p3f2.ff4e& ef4g4f f1
 3270 g(10)="@1v13o3L8y55,00p3r16f2.ff4e& ef4g4f @21v13o2r16ff<f
>ff<f>f
 3280 b(10)="@5o4v13L8y50,00p3f4.p1v13>gb-<dfa p3v13f2e2   v13p1
<c>fa<c&c2
 3290 d(10)="@5o4v13L8y52,24p3f4.p2v13>gb-<dfa p3v13b-2b-2 v13p2
<c>fa<c&c2
 3300 c(10)="@21v15o2L8p3>g16ggggggg <cccccccc r16 @5v10p3o4<c>f
a<c&c2
 3310 e(10)="@8v14p2L8gggggggg ggq8e4r4.v15p3e gp2v14gggggg
 3320 m_trk(2,a(10))
 3330 m_trk(3,b(10))
 3340 m_trk(4,c(10))
 3350 m_trk(5,d(10))
 3360 m_trk(6,e(10))
 3370 m_trk(7,f(10))
 3380 m_trk(8,g(10))
 3390 /*
 3400 /*
 3410 /*
 3420 m_trk(1,d4+s4+d8+d8+s4+d4+s8+d8+d4+s4+d4+s4+d8+d24+d24+d24
+s4+d4+s8+d8+d4+s4):m_trk(1,d4+s4+d8+d8+s4+d4+s8+d8+d4+s4+d4+s8+
d8)
 3430 m_trk(1,19+s6+s8+18)
 3440 m_trk(2,a(2))
 3450 m_trk(3,b(2))
 3460 m_trk(4,c(2))
 3470 m_trk(5,d(2))
 3480 m_trk(6,e(2))
 3490 m_trk(7,f(2))
 3500 m_trk(8,g(2))
 3510 m_trk(1,cr+s8+d8+d4+s4+d4+s8+d8+d4+s4+d4+s4+d4+s4+d4+s4+d8
+d8+s4)
 3520 m_trk(2,a(3))
 3530 m_trk(3,b(3))
 3540 m_trk(4,c(3))
 3550 m_trk(5,d(3))
 3560 m_trk(6,e(3))
 3570 /*f(11)="@14o3v08L2q5 f. c4 d.  a4 g1 ag
 3580 /*g(11)="@14o3v08L2q5 c.>a4 b-.<f4 e1 fe
 3590 m_trk(7,f(3))
 3600 m_trk(8,g(3))
 3610 /*
 3620 m_trk(1,d4+s4+d8+d8+s4+d4+s4+d4+s4+d4+s8+d8+d4+s4+d4+s4+w8
+18+s8+18)
 3630 m_trk(2,a(4))
 3640 b(12)="f4fe4f4a& ag4d4.ef& f2rfff f4fd4c4.
 3650 c(12)="f4y51,30fe4f4a& ag4d4.ef& f2rfff f4fd4c4.
 3660 /*f(12)="f1 d1 f1 f4.g8&g2
 3670 /*g(12)="d1 >b1< d1 c4.e8&e2
 3680 m_trk(3,b(12))
 3690 m_trk(4,c(12))
 3700 m_trk(5,d(4))
 3710 m_trk(6,e(4))
 3720 m_trk(7,f(4))
```

▶ つい先日秋葉原に電脳倶楽部を買いにいったんですよ（タケルで）。そうしたらパッケージの中に8月号プレゼントの4番が……。　　　白石　隆一(20)千葉県

```
 3730 m_trk(8,g(4))
 3740 /*
 3750 m_trk(2,a(5))
 3760 m_trk(1,cr+s8+d8+d4+s4+d4+s4+d8+d8+s4+d4+s4+d4+s4+d4+s8+d8
+d4+s4+d4+s4+d4+s4):
      m_trk(1,d4+s8+d8+d4+s4+d4+s8+d8+d4+s4+d8+s8+s8+s6+s6+s6+s6
+h6+h6+l8+s8)
 3770 b(13)="r4.cf4cd& d1 rggggab-a& a4.gg4.f f4fe4f4a& ag4d4e4f
& f2rfff fefg2r
 3780 c(13)="r4.cf4cd& d1 ry51,40v15ggggab-a& a4.gg4.f f4y51,30v
14fe4f4a& ag4d4e4f& f2rfff fefg2r
 3790 /*f(13)="f. c4 d. a4  g1 ag f1 d1   f1 f4.g2&g8
 3800 /*g(13)="c.>a4 b-.<f4 e1 fe d1 >b1<d1 c4.e2&e8
 3810 m_trk(3,b(13))
 3820 m_trk(4,c(13))
 3830 m_trk(5,d(5))
 3840 m_trk(6,e(5))
 3850 m_trk(7,f(5))
 3860 m_trk(8,g(5))
 3870 /*
 3880 m_trk(1,d4+s8+d8+w4+s4+d4+s8+d8+w4+s8+d8+d4+s8+d8+w4++s8+d
8+d4+s8+d8+w8+d6+d6+s8+d8)
 3890 m_trk(2,a(6))
 3900 b(14)="e4ff4g4f& fc4a2a a4gg4f4a& aggf4.rd
 3910 m_trk(3,b(14))
 3920 c(14)="e4ff4g4f& fc4a2a a4gg4f4a& aggf4.rd
 3930 m_trk(4,c(14))
 3940 m_trk(5,d(6))
 3950 m_trk(6,e(6))
 3960 m_trk(7,f(6))
 3970 m_trk(8,g(6))
 3980 /*
 3990 m_trk(1,d4+s8+d8+w8+d8+s4+d4+s8+d8+w8+d8+s4+d4+s8+d8+w8+d8
+s4):m_trk(1,d8+s8+s8+d8+s8+s8+s8+d8)
 4000 m_trk(2,a(7))
 4010 b(15)="a+16&b-8.b-b-4b-4a& ab-4<c4.>b-b- b-4ag4d4f& fefg2&
g
 4020 m_trk(3,b(15))
 4030 c(15)="v15a+16&b-8.b-b-.b-4a& ab-4<c4.>b-b- b-4ag4d4f& fef
g2&g
 4040 m_trk(4,c(15))
 4050 m_trk(5,d(7))
 4060 m_trk(6,e(7))
 4070 m_trk(7,f(7))
 4080 m_trk(8,g(7))
 4090 /*
 4100 m_trk(1,d4+s8+d8+w8+d8+s8+d8+d4+s4+d8+d8+s8+d8+d4+s8+s6+s6
+s8+h8+l8+l8+cy+w8+w8+w8+w8+cy)
 4110 m_trk(2,a(8))
 4120 b(16)="L4ef8f2&f8 ef8v14fgv15<c8& c2..>a8 v15b-ade8f8
 4130 m_trk(3,b(16))
 4140 c(16)="L4ef8f2&f8 ef8v14fgv15<c8& c2..>a8 v15b-ade8f8
 4150 m_trk(4,c(16))
 4160 m_trk(5,d(8))
 4170 m_trk(6,e(8))
 4180 m_trk(7,f(8))
 4190 m_trk(8,g(8))
 4200 /*
 4210 m_trk(1,d4+s4+d4+s4+d4+s4+d8+d8+s4+d4+s4+d8+d8+s4+d4+s4+d8
+d8+s4):m_trk(1,d4+s4+d8+d8+s4+d4+s4+d8+d8+s4)
 4220 m_trk(2,a(9))
 4230 b(17)="r4.L8cf4cd& d1 r4gggab-a&a4.gg4.f f4fe4f4a& ag4d&d4
ef&
 4240 m_trk(3,b(17))
 4250 c(17)="r4.L8cf4cd& d1 r4y51,40v15gggab-a&a4.gg4.f f4y51,30
v14fe4f4a& ag4d&d4ef&
 4260 m_trk(4,c(17))
 4270 m_trk(5,d(9))
 4280 m_trk(6,e(9))
 4290 f(17)="@14o3v12q7L1 f f e f2e2 f d
 4300 m_trk(7,f(17))
 4310 g(17)="@14o3v9q5L1 c d c c2c2 e >b<
 4320 m_trk(8,g(17))
 4330 /*
 4340 m_trk(1,d4+s4+d8+d8+s8+d8+d8+s8+s8+h8+l8+l8+l8+l8)
 4350 h(18)="L8                  "                      /*strings h
 4360 a(18)="o3>gga4b-b-<d4 ccddeeg4 "                  /*low strings
 4370 b(18)="f2.ff f4ef4fg4 f1
 4380 c(18)="f2.ff f4ef4fg4 f1
 4390 d(18)="v13p3f4.>gb-<dfa b-2b-2 "                  /*flee
 4400 e(18)="q4p2v15ev13eeeeeee v15ev13eeer2 v15e4"/*flee
 4410 f(18)="f1 f2g2 "                                  /*horn h
 4420 g(18)="d1 c2e2 "                                  /*horn l
 4430 m_trk(2,a(18))
 4440 m_trk(1,h(18))
 4450 m_trk(3,b(18))
 4460 m_trk(4,c(18))
 4470 m_trk(5,d(18))
 4480 m_trk(6,e(18))
 4490 m_trk(7,f(18))
 4500 m_trk(8,g(18))
 4510 /*m_play()
 4520 /*
 4530 a(19)="@12o2L2q7v14 >f1& f1 f1
 4540 b(19)="          r4@13o3@v126p3L8q8 b-ag2 r1
 4550 c(19)="      r8&r16@13o3@v126p3L8q8 ffe2  r1
 4560 d(19)="   r4r8y52,0@11p3v15o4q7L8 cf4c4      d1&dr4ggab-<c
 4570 e(19)="  r8@11y53,40r16   p3v13o4q7L8 cf4c4 d1&dr4ggab-<c
 4580 f(19)="@10o3v14L2q7 f1 r4r2p3f8b-8 g1
 4590 g(19)="@10o3v14L2q7 c1 r2r4p3d8f8  e1
 4600 m_trk(1,cy+"r2.r1r1")
 4610 m_trk(2,a(19))/*Lowstrings
 4620 m_trk(3,b(19))/*Flute
 4630 m_trk(4,c(19))/*Flute
 4640 m_trk(5,d(19))/*Strings
 4650 m_trk(6,e(19))/*Strings
 4660 m_trk(7,f(19))/*Horn
 4670 m_trk(8,g(19))/*Horn
 4680 /*
 4690 a(19)="<fe d.c4 >bg ga b<e-
 4700 b(19)="r1r1r1r1r1
 4710 c(19)="r1r1r1r1r1
 4720 d(19)=">a4gfeg<ce e4fec>afd >b<dag4.de f4efg4fg a-4b-4<c4d
-4
 4730 e(19)=">a4gfeg<ce e4fec>afd >b<dag4.de f4efg4fg a-4b-4<c4d
-4
 4740 f(19)="fe f.e4d1 b-<c c+e
 4750 g(19)="cc d.c4 >b1 b<c c+e
 4760 h(19)="r1r1r1r1r1
 4770 m_trk(1,h(19))
 4780 m_trk(2,a(19))
 4790 m_trk(3,b(19))
 4800 m_trk(4,c(19))
 4810 m_trk(5,d(19))
 4820 m_trk(6,e(19))
 4830 m_trk(7,f(19))
 4840 m_trk(8,g(19))
 4850 /*
 4860 a(20)="v11L8>g+4.b-<c4d+4 >g+1 g+b-<cd+g+4d4 >g+1 g2&g<d>b
-g
 4870 b(20)="@11y50,30o4v13g+1b-1g+1b-1 p1<d2&L8d>ga<dp3
 4880 c(20)="@4L16o3v10q6p3  r16  |:5cdefgab-<c>b-agfedc>b-<:|
 4890 d(20)="y52,20q8c1 d1 c1 d1 d2&d8>q7g8b-8<d8
 4900 e(20)="y53,40q8c1 d1 c1 d1 d2&d8>q7g8b-8<d8
 4910 f(20)="p1y54,44L8v11o2|:o3e-2a-ga-<c d4e-dd4>b-4:| f2&fdef
 4920 g(20)="p2y55,00L8v11o2|:o3e-2a-ga-<c d4e-dd4>b-4:| f2&fdef
 4930 h(20)="@4L16o3v12p2q5|:5cdefgab-<c>b-agfedc>b-<:|
 4940 m_trk(1,h(20))
 4950 m_trk(2,a(20))
 4960 m_trk(3,b(20))
 4970 m_trk(4,c(20))
 4980 m_trk(5,d(20))
 4990 m_trk(6,e(20))
 5000 m_trk(7,f(20))
 5010 m_trk(8,g(20))
 5020 /*
 5030 a(21)="t125<c2>b-2 v08a1& a1 b-1&b-1
 5040 b(21)="L4e-dc>g v8c1 @19o3v6p1L8cde-fgfe-g g4.gf2&f1p3
 5050 c(21)="cdefgab-<c>b-agfedc>b-< @11v08e-1o4 @19o3v5L8p3cde-
fgfe-g g4.gf2& f1
 5060 h(21)="cdefgab-<c>b-agfedc>b-< @11v10o4e-1&e-1 d1&d1
 5070 d(21)="e-4d4c4>g4 @v126g8&v14g8&v12g2.& v10g1 v8f1&f1
 5080 e(21)="e-4d4c4>g4 @v126g8&v14g8&v12g2.& v10g1 v8f1&f1
 5090 f(21)="v11g4f4e-4g4 v10e-4&v08e-2.r1 r2r8 p1   @13o3v13L8>
b-<cd L4v14fb-<cd
 5100 g(21)="v11g4f4e-4g4 v10e-4&v08e-2.r1 r2r8y55,24p2@13o3v13L
8>b-<cd v14L4fb-<cd
 5110 m_trk(1,h(21))
 5120 m_trk(2,a(21))
 5130 m_trk(3,b(21))
 5140 m_trk(4,c(21))
 5150 m_trk(5,d(21))
 5160 m_trk(6,e(21))
 5170 m_trk(7,f(21))
 5180 m_trk(8,g(21))
 5190 /*
 5200 a(22)="<v10c1&v12c2.@2v13L16q7o3>v15g&g-&f&f-L4 b-.<b-.>b
 a.<a>a8b-8a8
 5210 b(22)="r1 @11o3p1L16v14 gfga v15b-ab-<cdefg@v126ab-<cd @1v
15o3y50,00L4p3ef8fgf8& f8ca2a8
 5220 c(22)="r1 @11o3p2L16v14 gfga v15b-ab-<cdefg@v126ab-<cd @1v
14o3y51,40L4p3ef8fgf8& f8ca2a8
 5230 d(22)="v11g1 p2L16v14o3gfga v15b-ab-<cdefg@v126ab-<cd&d4 o
3L4v14p3>b-<c>b- a<c8f2&f8
 5240 e(22)="v11g1 p1L16v13o3gfga v14b-ab-<cdefg@v120ab-<cd&d4 o
3L4v12>b-<c>b- a<c8f2&f8
 5250 f(22)="v13p1cL8gfeceg L4@11o3v13<c&v13c&v14c&v15c4 @10q8o5
v11>a1 a1
 5260 g(22)="v13cp2L8gfeceg L4@11o3v13g&v13g&v14g&v15g4 @10q8o5v
11>d1 c1
 5270 m_trk(1,"r1"+w8+s8+s8+s6+s6+s8+s6+s6+s8+h8+d4+s8+d4+w8+s4+
d4+s8+d4+w8+s4)
 5280 m_trk(2,a(22))
 5290 m_trk(3,b(22))
 5300 m_trk(4,c(22))
 5310 m_trk(5,d(22))
 5320 m_trk(6,e(22))
 5330 m_trk(7,f(22))
 5340 m_trk(8,g(22))
 5350 /*                                      126
 5360 a(23)="o3e-.<e-.>e- d.<d.>b-8a8
 5370 b(23)="aggf8a8& a8g8g8f.d8d8
 5380 c(23)="aggf8a8& a8g8g8f.d8d8
 5390 d(23)="e-1 d1
 5400 e(23)="e-1 d1
 5410 f(23)="o4<c1>>d4e4f4<c4
 5420 g(23)="o4f1>d4e4f4<c4
 5430 m_trk(1,d4+s8+d4+w8+s4+d4+s8+s6+s6+s6+s6+h6+h6+l6+l6+l6+l6
)
 5440 m_trk(2,a(23))
 5450 m_trk(3,b(23))
 5460 m_trk(4,c(23))
 5470 m_trk(5,d(23))
 5480 m_trk(6,e(23))
 5490 m_trk(7,f(23))
 5500 m_trk(8,g(23))
 5510 /*
 5520 a(24)="g.<g>g8a8g8 g-.<g-.>g- f.<f.>f <c8cgc8<c8>c8
 5530 b(24)="b-b-8b-b-b-8 ab-8<c>b-b-8 b-a8gdf8& f8e8f8g2r8
 5540 c(24)="b-b-8b-b-b-8 ab-8<c>b-b-8 b-a8gdf8& f8e8f8g2r8
 5550 d(24)="b-1 <f+4.a8a8d8c8>b-8<c1 f4.g2&f8
 5560 e(24)="b-1 <f+4.a8a8d8c8>b-8<c1 f4.g2&f8
 5570 f(24)="f1<c1 >f1b-4.b-2&b-8
 5580 g(24)=">b-1<f+1> b-1< f4.e2&f8
 5590 m_trk(1,d4+s8+d4+d8+s4+d4+s8+d4+d8+s8+d8+d4+s8+d4+d8+s8+d8
+d4+s8+d4+d6+d6+s8+d8)
 5600 m_trk(2,a(24))
 5610 m_trk(3,b(24))
```

▶最近,友人のS氏が2Mバイトに増設した。これで「メルヘンメイズはこんなに面白いぞ」といじめることができなくなってしまった。まあ,今度は「MIDIはいいぞ」といじめてあげよう。
鈴木　武虎(17)愛知県

```
5620 m_trk(4,c(24))
5630 m_trk(5,d(24))
5640 m_trk(6,e(24))
5650 m_trk(7,f(24))
5660 m_trk(8,g(24))
5670 /*
5680 a(25)="o3>L8b-b-b-b-b-b-b-b- aaaaaaaa ggggg<dgd c1
5690 b(25)="e.f2r8 ef8fg<c8& c1 >b-ade8f8
5700 c(25)="e.f2r8 ef8fg<c8& c1 >b-ade8f8
5710 d(25)="r4.L8f4ef<c> a1 g1f1
5720 e(25)="r4.L8f4ef<c> a1 g1f1
5730 f(25)="a1 a1 g1 b-1
5740 g(25)="d1 c1 d1 e1
5750 m_trk(1,d4+s4+d8+d8+s4+d4+s4+d8+d8+s4+d4+s8+d8+s8+s6+s6+s8
+d8+d4+"r2.")
5760 m_trk(2,a(25))
5770 m_trk(3,b(25))
5780 m_trk(4,c(25))
5790 m_trk(5,d(25))
5800 m_trk(6,e(25))
5810 m_trk(7,f(25))
5820 m_trk(8,g(25))
5830 /*
5840 a(26)="@12o2v10L4f1& f1 eceg L2fe d.c4 >b1 g1 <c1 >f1&f1&f
2
5850 b(26)="L8r4.cf4cd& d1 r4.ggab-a& a4.gg4.f f4fe4f4a& ag4d4e
4f8& f2.f8f8f4e8f4g4f8 f1&f1&f2
5860 c(26)="L8r4.cf4cd& d1 y51,48v15r4.ggab-a& a4.gg4.f f4y51,4
0v14fe4f4a& ag4d4e4f8& f2.f8f8f4e8f4g4f8 f1&f1&f2
5870 d(26)="o3L8v8f2&v10f4&f4&v12 f4b-4agfe& e1 <c2e2 f2&f>a<da
g1g4>b-4<c4d4f2g2 q8f1&f1&f2
5880 e(26)="o3L8v8f2&v10f2&v12 f4b-4agfe& e1 <c2e2 f2&f>a<da g1
g4>b-4<c4d4f2g2 q8f1&f1&f2
5890 f(26)="@11o3v8L8q7y54,40 c2&v10c2&v12c4p1b-4agfp3c& c1 a2<
c2 d2&d>p1a<da p3d1 d4p1L4>b-<cdp3c2e2 >q8f1&f1&f2
5900 g(26)="@11o3v8L8q7y55,32 r16 c2&v10c2&v12c4p2b-4agfp3c& c1
a2<c2 d2&d>p2a<da p3d1 d4p2L4>b-<cdp2c2e2 >q8f1&f1&f2
5910 h(26)="@11o3v8L8q7y48,32 >a2&v10a2&v12a4 <b-4agf>g& g1 <f2
g2a2&aa<da f1> b-4b-4<c4d4 >g2<c2 q8f1&f2
5920 m_trk(1,h(26))
5930 m_trk(2,a(26))
5940 m_trk(3,b(26))
5950 m_trk(4,c(26))
5960 m_trk(5,d(26))
5970 m_trk(6,e(26))
5980 m_trk(7,f(26))
5990 m_trk(8,g(26))
6000 m_play()
```

日本音楽著作権協会(出)許諾第9170916-101号

リスト2 WHITE MANE

```
 '
 0 ' T-SQUARE  NATURAL
30 ' 「 WHITE MANE 」  Music by Hirotaka Izumi
40 ' Programed by T.Abe
50 '
60 DIM a$(41),p(30):GOSUB 1810
70 PLAY 0:PLAY "t90":GOTO 120
80 LABEL"set"
90 READ a:IF a=255 THEN PLAY ":";:RETURN
100 PLAY a$(a);:GOTO90
110 '
120 ' track 1
130 a$(0)="r1r1r1r1"
140 a$(1)="i20L8v15o4k5"
150 a$(2)="b4&b16g>d16<a4&a12>c12<b12> c8.g16&g12f+12e12ef+16d<b
16>c16d16 de16<gab>dd<d16g16b16> c<b16>c<bbaa16&a12g12a12"
160 a$(3)="b8.>d8.gf+8.a16&a12g12f+12 g8.>c<ba16b16f+d<b8. >c8.<
b8.ga2 g1"
170 a$(4)="L16r8.dgaba4>d8c8<b>c8<bg8e8e8d8.&d4 r8.e8f+8gg8.d8<b
>cd c8<b>c8g8d8d8.r4"
180 a$(5)="r8.dgaba4.>dc<b>c8.<b8g8a8.b8.d8.e8g8.ge>d&d4.d<b>d8.
d&d2.<L8"
190 a$(7)="L16r8.g8f+8ee8f+d8<a8b> c8<b>d+rerf+a8.g8.ar8 b8.>d8.
e8.d8.<b8.rg8.&g8g8a8.a&a4"
200 a$(8)="r8.d8g8>dd8c<b8a8g& g4&gdgbb8>c<a8g8f+& f+4&f+g8aa8bg
8d8e& e4&ea8gg8f+g8a8."
210 a$(9)="g8rd8g8>dd8c<b8a8g& g4&gb>dgg8f+f+8e8d& d4&d12<b12>d1
2<g4&gega bag4.&g2& g4r4r8.ergra"
220 a$(10)="bag4.&g2& g4r8eerdegrega bag4.&g2& ga8.a8.a8.a8rega
230 a$(11)="L8<g2>>rd4.>c4.<g4.r4 >d4.<a4.r4 g4.erg4. <b2>g>d4.
c4.e4.r4 f+4.dr <a4."
240 a$(12)="g2c>c<gd b1 g2r>c4.r4.d4a4. g4.e4c8.e8. d8.<b16&b4 g
4&g16>d8. c2r<g4. f+8.a4a8.d4. r4.&r16>e32d32<g2<"
250 a$(13)="b8.>d16&d4<a4&a16>d8.< g4&g16>c8.<b4&b16>d8. c8.e16&
e4<b4&b16>d8.< a4&a16e8. g8.a16&a4"
260 a$(14)="b4&b16>d8.<a4&a16>d8.< g4&g16>c8.<b8.>d16&d4< g4&g16
g8.>c2< g2r2"
270 a$(15)="r2.r16a8. g4&g16>e8.<f+8.f+16&f+4 e8.c16&c4d8.<b16&b
4> e8.c16&c4 g4a4"
280 a$(16)="b8.>d16&d4<a8.>d16&d4< g8.>c16&c4<f+8.b16&b4 e4&e16e
8.g4a4g2r2"
290 a$(17)="gf+e8.f+g>d8<f+gf+8g8. c4&cg8.a8.>d8.<a8"
300 a$(19)="g8.f+8.<g4>d8<d4 g8<b>c8g8c8cde8g8. <b8.>e8.e4b8.e8.
"
310 a$(20)="<a8.>d8.>a8.<d4<b8.&b8.>>f+&f+4<b8.>e8f+8.< b8.e8b8.
&b4b8>d8< b8.b&b4.>e8.<b8."
320 a$(21)="a8.d8>a8.f+8.d8.<b8& b8.>f+&f+4.<ab8>d8.< b2&b8>d8.<
b8. g8.>g&g4.b8.g8."
330 a$(29)="L16gf+e8.f+g>d8<f+gf+8g4 >f+8g8b8>g8d8e8<b8>dc8.<b8.
g4.&g<b>cgf+8.a8.d4.g12f+12a12"
340 a$(30)="r8.gf+e16ef+16d<ab16> c<b16>f+16&f+12<b12a12a8.g16&g
4 r8.egf+16ga16b>dd8. e16<bga16&a2>"
350 a$(41)="`10\"
360 DATA 0,1,2,3,4,5,30,7,8,9,10,17,11,12,13,14,15,16,1,30,7,8,9
,10,29,19,20,41,21,255
370 GOSUB "set"
380 ' track 2
390 a$(1)="i20L8v15o4k10"
400 a$(2)="d4&d16g>d16<d4&d12>c12<b12 g8.>g16&g12f+12e12<f+>f+16
d<b16>c16d16 <g>e16<gadagd16g16b16 ed16eddcc16&c12g12a12"
410 a$(3)="b8.>d8.<ba8.>d16&d12g12f+12 c8.e8dc16d16<bf+d8. e8.d8
.<b>c2 g1"
420 a$(4)="L16r8.dgaba4>d8c8<b>c8<bg8e8e8d8.&d4 r8.e8f+8gg8.d8<b
>cd c8<b>c8g8<b8a8.r4>"
430 a$(5)="r8.dgaba4.>dc<b e8.d8<b8>e8.f+8.d8.e8g8.gea&a4.aba8.g
&g2."
440 a$(6)="i7o5v14L8k10"
450 a$(7)="i20L8v15o5k10"
460 a$(8)="r1e4.c4.r4 f+4.d4.r4 c4.<g4>c4. <d2>g2 e4.a4.r4 a4.f+
4d4."
470 a$(9)="c2c>c<gd d1 g2r>c4. r4.<f+4>d4. c4.<g4e8.>g8. g8.b16&
b4 g4&g16g8. e2rg4."
480 a$(10)="<a8.>d4d8.<f+4. r1"
490 a$(11)="g2d4&d16d8.e4&e16g8.f+4&f+16b8. g8.>c16&c4<b4&b16g8.
 e4&e16c8.e8.e16&e4"
500 a$(12)="d4&d16d8.d4&d16d8. c4&c16e8.d8.f+16&f+4 c4&c16c8.e2
<d8.e16&e4d8.e16&e4"
510 a$(13)="d4&d16d8.d4>r16d8. c4&c16g8.<b8.b16&b4 g8.g16&g4g8.g
16&g4 g8.g16&g4>c4c4"
520 a$(14)="d8.d16&d4d8.d16&d4 c8.c16&c4<b8.b16&b4 a4&a16a8.>c4c
4 <e8.d16&d4e8.d16&d4>"
530 a$(24)="gf+e8&e2.&e2r8.e@4d@4<b@4a8.ga8g>e8d8<a8bga&a4 r8L32
a&>d&<ag& a16a&>d& <a&g&a16 a>d<ag a&g&a&a&a16r8b&a&L16"
540 a$(25)="b8.>d32&<b32&>d8.e32&d32gaa#b>d<ba#a g8r8.g8.r2 r16L
32b>dc+c<ba#ag#gd#eff#ga16a16&agede8g16a16 r4gabgaagedged<b>ed<b
a16.>d@3&<b@3&a16ga"
550 a$(26)="r2r16.e@2g@2a@2>d16<a@4b@4>d@4 e8 <b@4>d@4e@4g@4a@8"
560 a$(27)="bagea16g16g8e16r4r16L@8f+d+ef+gaL32a#b>dc#c<ba#ag#gf
+16r8r8.L16ega>d<b"
570 a$(28)="b8aag8a8r4.b>de8d8d<b>dd8.r8.<ab>de4d<b>dd&d4r<b>de
g2L32ed<ba#aged<a#bgf+e8"
580 a$(31)="L16r8.g8f+8ee8f+d8<a8b> c8<b>d+rerf+a8.g8.ar8 b8.>d8
.e8.d8.<b8.a2.r4"
590 a$(32)="r8.d8g8>dd8c<b8a8g8.r8.dgbb8>c<a8g8f+8.r8.g8aa8bg8d8
e8.r8.a8gg8f+g8a8."
600 a$(33)="g8rd8g8>dd8c<b8a8g8.r8.b>dgg8f+f+8e8d8.r8&r12<b12>d1
2<g4rega bag4.&g2&"
610 a$(34)="g4r4r8.ergra bag4.&g2&g4r8eerdegrega bag4.&g2&g4r2 r
ega"
620 a$(40)="gf+e4.&e2&e4r2."
630 DATA 0,1,2,3,4,5,6,30,31,32,33,34,40,7,8,9,10,11,12,13,14,6,
30,31,32,33,34,24,25,26,27,28,255
640 GOSUB "set"
650 ' track 3
660 a$(1)="i9o3v11L8"
670 a$(2)="g2f+2e2d2c2<b2a2>d4<d4"
680 b$="gr16g16&g4 f+r16f+16&f+4 er16e16&e4 dr16d8.>d16&c16"
690 a$(3)=b$+"cr16c8.c16c+16dr16d<a16>d <gr16g16&g4 gr16g16&g4"
700 a$(4)=b$+"cr16c16&c4 <br16b16&b4 ar16a16&a4>dr16d<a16>d<"
710 a$(5)=b$+"cr16c8.c16c+16dr16d<a16>d< gr16g16&g4gr16g8.>d"
720 b$="cr16c<g16>cdr16d<a16>d"
730 a$(6)=b$+"d+r16d+<b16>d+er16e<b16e ar16ae16abr16bg16b >cr16c
<g16>cdr16d<a16>d"
740 a$(7)=b$+"d+8.<b16>r16c+16r16d16e8.d8.<g> r16c8.<b8.a8.b8.>c
8. r16d8.d4d8.d<a16>d<"
750 a$(8)="L16g8rgg8rga8raa8ra b8rbb8rb>c8rcc8cc+ d8rdd+8rd+e8re
d8rd c+8rc+c+8<a>c+d8rdd<a>d8<"
760 a$(9)="g8rgg8rga8raa8ra b8rbb8rb>c+8rc+c8rc< b8rbb8f+be8re>e
8d8 c8.<gg8.>cc8.<gg8g&a"
770 a$(10)="a8.>ee8.<aa8.>ee8d8 c8.<gg8.>cc8.<gg8.a a8.>ee8.<aa8
.>ee8d8 c8.<gg8.>cc8.<gg8>c8 rd8.d8.d8.drr4"
780 a$(11)="e8.<bb8.>ee8.<bb8.>e e8.cc8eff+8.d8d"
790 a$(12)="i31v14L16o2g@4&a@20"
800 a$(13)="g2&g8.dgab>d d&ccd&e2&eef+g e@6&f+@30d2<br>crd d&ccc
8<b8b&>c<g8.&g4"
810 a$(14)="r4r>f+&gf+dec&dc&<b>cd< b@4&>c@20<ba8g&f+e8>c8.&c8.<
e f+8f+>d4d32&e32d8.c&<b>cd<b32&>c32& <bagf+8g8&e8c8.&c8r>c&"
820 a$(15)="dc&<b>g4f+gaf+d4g abge4ab>c<ae4b >cd<bg8f+8g8a8b8>c8
c& dd8<b8d8.&d8.f+gab>c"
830 a$(16)="c&dd&<b8g&ag&g4L24b&>c&d<b&>c&dL16< bgaf+32&g32&f+ed
c4>c&<bgaf+32&g32 f+8f+e8e&dd4g&f+dec32&d32 c8c<b8g8e8c8. r>g8."
840 a$(17)="i9o3v11L16"
850 b$="g8rg&g4f+8rf+&f+4"
860 a$(18)=b$+"e8re8.e8 d8rd8.>d&c c8rcc8rc<b8rb8.rb a8ra8.ra>d8
rd8<a>d8<"
870 a$(19)=b$+"e8re4ed8rd8.>d&c c8rcc8cc+d8rd8<a>d8< g8rg&g4g8rg
8.d8"
880 a$(20)=b$+"e8re4ed8rd8.>d&c c8rc8.rc<b8rb8.rb a8ra8b>c<a>d8r
d8<a>d8<"
890 a$(21)="g8rg8.rgf+8rf+8.rf+ e8re8.red8rd8.dc c8rc8.cc+d8rd8<
a>d8 <g8rg8.rdg8rg8.>c+@4&d@20L8"
900 a$(22)="e8.<bb8.>ee8.<bb8.>c c8.gg8.cc8.gg8.<a a8.>ee8.<aa8.
>ee8.<b b8.>f+f+8.<bb8.>f+f+8.<b>"
910 DATA 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21
,6,7,8,9,10,22,22,22,22,22,255
920 GOSUB "set"
930 ' track 4
940 a$(1)="o6v13r1r8.i32g16r8c4.r4 d16d16r8 c4r4r4 i1v13s3,1,0,1
=3c1=0"
950 a$(2)="i22o4v12L16"
960 b$="gd<b>g8gdgf+ada8adf+ ecge8gcedf+<b>f+8f+<b>d ecge8cge"
970 a$(3)=b$+"<b>dgd8dgd ed8e8e8d8<a>ed8f+ad"
980 a$(4)=b$+"aceg8df+a gd<bg8>gecgd<bg8>gec"
990 b$="da"+RIGHT$(b$,51)
1000 a$(5)=b$+"<b>dgd8dgd ed8e8e8d8<a>ed8f+ad"
1010 a$(6)=b$+"dgcd8f+ad ceag8gcgcge<b8>gdg"
1020 b$="egce8gcedf+<a>d8f+<a>d"
1030 a$(7)=b$+"d+f+<a>f+8f+<a>deg<b>e8ege cege8gce<b>dgd8g<b>d c
```

▶ IOCS用FONT200書体のところにあった“A QUICK BROWN FOX JUMPS OVER THE LAZY DOG”っていう文にはまいりました。「何だこの英文は？」と思っていたらちゃんとA～Zまでの文字が入っていたのか……。　飯村　直樹(16)静岡県

```
ege8egef+df+d8f+ad"
1040 a$(8)=b$+"i20L16v12o3 r4rbrb>c8.f8.er8 e8.d8.c8.d8.e8. re8.
&e8e8f+8.f+&f+4"
1050 a$(9)="i19o3L4v11d1&dgf+edc<bag2.r4>d1&dg2f+&f+2g2e1&e1&e1&
e2f+2g1r1"
1060 a$(10)="i22o4L16r8<b>deg8d8.b8.<b8.>c4&c16g8.a8.>d8.<a8"
1070 a$(11)="<b1>e1f+1e2.c8<g8 dgb>d&d4g2 e1f+1e2 >e8.<g8.e8<b1>
e1f+1e2&e8e8c8<gab2&b8.>d8.<b>ce1 f+1 e8.g8.e8>c4&c8."
1080 a$(12)="i22o5v12L8e8.<b8.g&g2 c8.dbg16&g2 <a8.>c<b>e16&e2 <
b8.>ded16&d2"
1090 a$(13)="r4.g4.r16b8. r<b16>cgc16&c2 <b16>cege16&e2 <b8.>ded
16&d2"
1100 a$(14)="e8.gf+b16&b2 cd8.bg16&g2 <a8.>c8.<b16>e16&e2"
1110 a$(15)="<b8.>ded4f+16&f+4 <b8.>d16r8.<b16>d8.d16&d4 c8.d16r
16c16r16c16d8.d16&d4 c8.<a16&a4>c2"
1120 a$(35)="i8o4L16v15a32&g32&"
1130 a$(36)="f+def+g&f+gabgab>c&<b>cd edef+g&f+ga a&b8f+8.d8 e8r
<b&>cegabg>cc+d d@4& d@20d8&< agg8gb>df+8ge&f+e&dd&"
1140 a$(37)="d8.g8f+8ed8.e&f+12e12&d12 c8.d&ed8c<b8.>c&<bage gef
+g&f+ed<b>d<b>c<abgag a8&ga8b8g&a8&gg8&df+a"
1150 a$(38)="g8.a>dgf+8.df+a>d<f+a>dc8.<eg>c<b8.df+ab&ab>d c8.<g
>ced8.<b>dga&gab b8>c<a&ba8gf+<g+&a@8>c@8e@8ag&f+e"
1160 a$(39)="d8<_5b~5>g8a8bf+8_d~f+8g8a e8_c~e8f+8gd8r<b&agf+&d
e8&de8g8ab8&>c<b8.a8 g8.a8.b8>d8.g&g4"
1170 DATA 1,2,3,4,5,6,7,8,9,10,11,35,36,37,38,39,2,7,8,9,12,13,1
4,15,255
1180 GOSUB "set"
1190 ' track 5
1200 a$(1)="i20L8v12o3k5"
1210 a$(2)="g8.>d16&d4<f+a16>d16&d4< e>c4.<b2> c2d2< a2d2"
1220 a$(3)=">d2d2 c2<b2> c2<g2 b8.>c16&c4< b8.>c16&c4<"
1230 a$(4)="b8.b16&b4>d8.d16&d4 c8.c16&c4<b8.b16&b4 g8.g16&g4g8.
g16&g4 g8.g4f+f+8.r4"
1240 a$(5)="b8.b16&b4a8.a16&a4 g8.g16&g4f+8.f+16&f+4 g8.g16&g4>c
8.c16&c4 c8.<b16&b4>c8.<b16&b4>"
1250 a$(6)="c8.c16&c4<a8.a16&a4 f+8.f+16&f+4g8.e16&e4 c8.c16&c4d
8.d16&d4 g8.g16&g4f+8.f+16&f+4"
1260 a$(7)="g8.g16&g4a8.a16&a4 f+8.f+16r16f+16r16f+16f+8.d8.g16r
 g8.g8.g8.g8.g8. r16d8.&ddd8.d16&d4"
1270 a$(8)="b8.b16&b4>c8.c16&c4 d8.d16&d4e8.e16&e4 d8.d16c8.c16<
b8.b16b8.b16 a8.a16&a4>c4c4<"
1280 a$(9)="b4r4>c4r4 d4r4<b4b4 a4r4b4r4 r4.L16>>>de<b>d<abgaegd
e<b>d<abgae4r4L8"
1290 a$(10)="r4.r@12>b@4>c@4d@4g2r1<r2 d16d16d16dd16d<r16e8.e8.e
8.e8r4k10"
1300 a$(11)="d2r4r16d8. c4&c16d8.d4&d16f+8. e8.g16&g4d4&d16d8. c
4&c16<a8.>c8.c16&c4"
1310 a$(12)="g4&g16g8.a4&a16a8. e4&e16e8.f+8.f+16&f+4 e4&e16e8.g
2 <b8.>c16&c4<b8.>c16&c4<"
1320 a$(13)="b4&b16b8.a4>r16f+8. g4&g16>c8.<f+8.f+16&f+4 c8.c16&
c4d8.<b16&b4>c8.c16&c4e4e4"
1330 a$(14)="g8.g16&g4a8.a16&a4 g8.g16&g4f+8.f+16&f+4 c4&c16c8.e
4e4 c8.<b16&b4>c8.<b16&b4>"
1340 a$(15)="k5r1r2<f+2>"
1350 a$(16)="g8.f+8.<g4>d8<d4d8<b>c8g8c8cde8g8.<b8.>e8.e4c8.e8."
1360 a$(17)="<a8.>d8.>d8.<d4<b8.&b8.>>d&d4<b8.>e8f+8.<b8.e8b8.&b
4b8>d8<b8.g16&g4.g8.e8."
1370 a$(18)="d8.d8>d8.<a8.f+8.b4&b>d&d4.<ab8b8. g2&g8b8.g8. e8.>
e&e4.e8.e8."
1380 DATA 0,1,2,3,4,5,6,7,8,9,10,15,0,0,0,0,11,12,13,14,6,7,8,9,
10,29,16,17,18,255
1390 GOSUB "set"
1400 ' track 6
1410 a$(1)="i20L8v12o3k10"
1420 a$(2)="<g8.>>d16&d4<f+a16>d16&d4< e>c4.<d2 c2<b2 a2>d2"
1430 a$(3)="g2f+2 e2d2 g2d2 d8.e16&e4 d8.e16&e4"
1440 a$(4)="g8.g16&g4a8.a16&a4 g8.g16&g4f+8.f+16&f+4 e8.e16&e4d8
.d16&d4 <a8.a4>dd8.r4"
1450 a$(5)="g8.g16&g4f+8.f+16&f+4 e8.e16&e4d8.d16&d4 c8.c16&c4g8
.g16&g4 e8.d16&d4e8.d16&d4"
1460 a$(6)="g8.g16&g4d8.d16&d4 d+8.d+16&d+4e8.e16&e4 <a8.a16&a4b
8.b16&b4> c8.c16&c4d8.d16&d4"
1470 a$(7)="c8.c16&c4d8.d16&d4 d+8.<b16>r16c+16r16d+16e8.>c8.<b1
6r c8.<b8.a8.b8.>c8. r16>c8.&ccd8.d16&d4<"
1480 a$(8)="g8.g16&g4g8.g16&g4 g8.g16&g4g8.g16&g4 d8.d16d+8.d+16
e8.e16d8.d16 c+8.c+16&c+4d4d4 d4r4e4r4 g4r4c+4c4 <b4r4>e4r4"
1490 a$(9)="i22L16o4"
1500 a$(10)="r8cdeg8b&b2 r8<b>ccege&e2"
1510 a$(11)="r1r1"
1520 a$(12)="i24v15o4r1r4.p1ddr2r1r1r4.p2ddr2r1r1r2r4.p1ddr1r1r1
r4.p2ddr2r1r1r4.p1ddr2r2.r8.p3"
1530 a$(13)="v13k15"
1540 a$(14)="i20o3r2>bbbb8ag8<r>c8.c8.c8.c8r4"
1550 a$(15)="v10r2.<e4c4r2.g4r2.f+4r2.<e8.>>e16&e4r2<c4r2.<a8.>a
16&a4r2<b2r2 e8.>>e16&e4r2<c2r2<a8.>a16&a4r2"
1560 a$(36)=LEFT$(a$(36),66)+"<a@4&b@20a8"+RIGHT$(a$(36),21)
1570 DATA 0,1,2,3,4,5,6,7,8,9,10,10,14,11,12,35,13,36,37,38,39,1
,6,7,8,9,10,10,14,0,15,255
1580 GOSUB "set"
1590 ' track 7
1600 a$(1)="i23L16o7v12"
1610 a$(2)="a_3ggg~3"
1620 a$(3)=a$(2)+a$(2)+a$(2)+a$(2)
1630 a$(4)=a$(3)+a$(3)+a$(3)+a$(3)
1640 a$(5)="v10"
1650 a$(6)="v13"
1660 a$(7)=a$(2)+a$(2)+"a_3gg~3a32a32"+a$(2)+a$(3)+a$(3)+a$(3)
1670 a$(8)=a$(2)+a$(2)+"a_3gg~3a32a32"+a$(2)+a$(3)+"raggaggaggag
gagg"+a$(3)
1680 a$(9)="r1"
1690 DATA 1,5,4,6,4,4,4,4,7,8,4,4,4,9,3,3,4,4,4,4,4,4,4,4,7,8,4,
4,4,9,4,4,4,4,255
1700 GOSUB "set"
1710 ' track 8
1720 a$(1)="i19o3L8"
1730 a$(2)="v5d1&v7d1&v9d1&v11d1&"
1740 a$(3)="d1&d1&d4c4<b2&b1>d1&d1&d4c4<b4a4g1"
1750 a$(4)="i1o5v16L8k5"
1760 a$(5)="v9r1c1d1c1<b1>c1d1c1<b1>c1d1c1d1c2&c8<g4.f+4a2>d4&d1
&d1&d1&d4c1r2.d1&d1&d4c4<b4a4g1&g4r2.r1r1r1"
1770 DATA 1,2,3,0,0,4,30,31,32,33,34,40,1,5,0,4,30,31,32,33,34,2
4,25,26,27,28,255
1780 GOSUB "set"
1790 PLAY ""
1800 END
1810 MEM$(&HB190,36)=HEXCHR$("FA 50 61 08 51 02 1B 27 25 00 5D 5
D 5D 50 04 0F 1F 1F 00 00 00 00 16 58 03 0A 00 80 80 00 00 C8 94
 00 02 00") ' Lead 1
1820 MEM$(&HB268,36)=HEXCHR$("BA 50 01 03 01 01 1A 2D 1E 00 1F 1
7 17 15 0E 0C 00 00 00 00 00 00 02 F2 02 07 00 00 00 00 D0 C8 94
 00 02 00") ' Lead 2
1830 MEM$(&HB28C,36)=HEXCHR$("FC 00 62 61 33 31 19 08 19 08 12 1
9 12 19 0A 0A 0A 0A 02 02 02 02 F3 3A F3 3A 00 00 00 00 00 C8 94
 00 02 00") ' E.Guitar
1840 MEM$(&HB2B0,36)=HEXCHR$("C3 00 65 40 40 30 2F 17 21 00 1F 1
F 1F 1C 0C 00 00 80 00 00 04 06 FA 0A 06 08 06 04 00 00 00 C8 80
 00 02 00") ' E.Bass
1850 MEM$(&HB418,36)=HEXCHR$("C4 00 34 34 74 74 14 00 14 00 54 8
A 54 8A 00 80 00 80 01 01 01 01 F3 FA F3 FA 00 00 00 00 E8 C8 94
 00 02 00") ' Strings
1860 MEM$(&HB43C,36)=HEXCHR$("FA 00 75 25 41 71 29 28 27 00 D3 D
3 93 D3 02 02 02 82 01 01 01 01 34 34 34 34 00 00 00 00 00 C8 94
 00 02 00") ' A.Piano
1870 MEM$(&HB484,36)=HEXCHR$("F9 00 6C 33 43 41 0B 21 20 00 9F 5
F 1C 1F 16 06 06 08 08 00 00 00 76 36 F6 F8 00 0A 0A 00 F4 C8 94
 00 02 00") ' A.Guitar
1880 MEM$(&HB4A8,36)=HEXCHR$("F8 00 0F 0F 0F 0F 01 02 00 09 1F 1
F 1F 10 00 00 00 92 C0 00 00 00 02 02 02 F9 00 00 00 00 18 C8 94
 00 02 00") ' Shaker
1890 MEM$(&HB4CC,36)=HEXCHR$("FC 00 06 08 0F 79 21 00 2F 0B 1E 1
C 1C 58 1A 94 14 93 80 00 00 00 FA FA FA F9 00 00 00 00 E8 00 80
 00 00 00") ' W.Block
1900 MEM$(&HB5C8,36)=HEXCHR$("C8 00 71 13 71 01 32 2D 19 00 55 5
F 95 1F 0A 87 05 81 0F 0F 0F 0F F4 38 F4 F8 00 00 80 00 F4 C8 80
 00 02 00") ' W.Bass
1910 MEM$(&HB5EC,36)=HEXCHR$("C4 70 02 02 22 72 39 0A 39 0A 11 1
0 0F 11 00 92 00 91 00 00 00 00 0A FA 0A FA 00 00 00 00 18 D2 F1
 00 02 80") ' Bird
1920 RETURN
```

（善）のゲームミュージックでバビンチョ

コンニーチハ，メルシーボクー。前置きはこれぐらいにして，さっそくいってみましょう。

●STRIKE FIGHTER/SEGA S.S.T.
CD：PCCB-00067　ポニーキャニオン　1,500円（税別）

「STRIKE FIGHTER」っていうのはいわゆる「G-LOC」のニューバージョンっていやつです。ゲームがニューバージョンなら曲のほうも……というわけで「STRIKE FIGHTER」の筐体からは「G-LOC」とはまったく別の曲が鳴っていました。さて，CDの中身ですがアレンジが2曲とオリジナルサウンドが10曲というごく普通の構成。でも相変わらずアレンジが凄い出来。1曲目の「BORN OUT」のオルガンソロがかっこいい。S.S.T.のアレンジバージョンはほかのメーカーのGMのそれよりも格段に編曲がしっかりしている感じを受けますね。

・オリジナルサウンドのほうはいつものセガセガしい調子です……。　　お勧め度　8

●THUNDER ZONE/DATA EASTGAMADELIC
CD：PCCB-00068　ポニーキャニオン　1,500円（税別）

いままでGMでサンプリングギターが鳴っているというとせいぜいコードを押さえている程度でした。がこの「THUNDER ZONE」は違います。ギターがリードパートでギュインギュイン鳴っているんです！　オリジナルサウンドでですよ。こいつは凄い。というわけで曲のほうもヘビメタあり，ブラックありとなかなか多彩な内容。ぜひご一聴あれ。

・私はこのゲーム，写真でしか見たことがないんですけど……。　　お勧め度　8

●COTTON/SEGA・SUCCESS　CD：PCCB-00069
ポニーキャニオン　1,500円（税別）

皆さんご存じのミーハーシューティング「COTTON」のBGM集です。ゲーム内容とは裏腹に曲のほうはFM音源数声だけで地味に鳴っています。CDの最後に入っているS.E.& VOICEコレクションをお目当てに買うのもいいかもしれません。

・「COTTON」ってゲームそのものは結構面白いんだよね。X68000に移植されないかな。

お勧め度　5

シミュレーションプログラミング入門 最終回

「株式」学問のススメ

Kamon Masato 華門 真人

長らく連載をお休みさせてもらって，復活したとたんに最終回。毎回楽しみにしていた方々には，たいへんもうしわけなく思います。「株式」シミュレーションの話で連載の最後をしめくくりますが，またいずれ違った形でお会いしましょう。

戻ってきたら，夏だった。そんな感じだ。ついこの間までは冬だったのに。寒さの中を温泉につかって，はしゃいでいたっけ。それがいまや，空気はすっかり夏の気配を帯びている。

とても個人的な話になるけれど，僕が四季の中でいちばん好きなのが春だ。暑くもなく，寒くもなく。花は厳しい寒さが去ったことを喜んでか，競って自己主張にいそしんでいる。

緑はしだいにその青さを増していく。ふと気がつくと，木々は鮮烈なまでに青い衣をまとっている。

もちろん，夏だって好きだ。ジェットスキーでチン（沈没）しても寒くなんかない。むしろ，気持ちいいぐらいだ。何をしても楽しい季節，それが夏だ。

秋は秋で，その涼しさ，そして空気に漂う寂寥感がなんともいえない雰囲気をかもし出している。そしていうまでもなく，冬には雪と，スキーと，そして温泉がある。

結局，僕はすべての季節が好きなのだ。でも，そんな中であえて春がいちばん好きといい切れるだけの理由。それは，そのはかなさにあるのだろう。

1年の中でもほんの一瞬だけ，自然がその躍動感をあらわにするとき。それが春だ。生き物はまるで生まれたての赤ん坊のように，生命の喜びを歌いあげる。

イタリアでは，まさに春真っ盛りだった。涼しげな緑を彩る，数多くの花々。多くは名も知らぬ花々だ。ヴィヴィッドな黄色，淡い桃色，清楚な白。皆，思いおもいの装いで，生きることの楽しさを訴えかけてくる。

でも，花の命は短いものだ。もはや季節は一目散に夏へと流れつつある。誰もその流れを押し止めることはできない。その流れの中で，花は無情にもその命を散らしていく。

こうして，春も去った。そして，いまは夏，春に生まれいでた新しい生命が，次なるステップへと確実に歩み始める時期だ。

花になぞらえるわけではないけれど，この「シミュレーションプログラミング入門」も今回をもっていったん幕を閉じることになった。

なぜなら，夏こそ飛躍のときだからだ。

review

半年以上にわたって，シミュレーションについて学んできた。すべては，渋滞のメカニズムをシミュレーションによって解析しよう，という試みから始まった。

もちろん，すぐさまそんなシミュレーションを実現できるはずはなかったのだけれども，まずは「離散変化型」モデルを導入することによって，解決の糸口を見い出すことができた。

こうして出来上がったのが料金所モデルだ。クルマの流入速度が，料金所の処理速度を越えると，とたんに大渋滞になるということがおわかりいただけたと思う。その離散変化型をもっとうまく使ってやれば，一般の渋滞シミュレーションも当然可能になってくるはずだ。

そのための手段として，GPSSと呼ばれる専用言語があることもご紹介した。

記憶力のいい方なら覚えていらっしゃるかもしれないが，一般的なシミュレーションの場合，

・離散変化型
・連続変化型

という，2通りの方法がある。いうまでもなく，「料金所モデル」は前者に属する。

これでは片手落ちだ，ということで連続変化型に挑戦してみたのが，PID制御による温水器モデルだ。これもサーキットシミュレーションという大きな夢を持ちながら，とりあえずは手近なところで，と温水器に挑戦してみたわけだ。お手軽版とはいえ，連続変化型ならではの，モデルが時間とともに刻々と変化していく様子がおわかりいただけたと思う。

さらに，連続変化型の応用として，ファジィ制御のシミュレートにも挑戦してみたのは，ご記憶に新しいところだろう。

investment

今回は少々毛色の変わったシミュレーションをご紹介することにしよう。いままでの渋滞シミュレーションだの，温水器シミュレーションだのといったものとは，少し肌合いを別にする。むしろ皆さんおなじみのシミュレーションゲームに近いものだ。ゲーム感覚で株式の勉強をしようという，その名も「投資シミュレーション」。

念のためいっておくとひと口に投資といっても，そうそう単純なものじゃあない。もちろんいわゆる「株」がいちばん有名ではあるけれども，そのほかにも，中国ファンドなどの投資信託，社債，そして社債は社債でも株式的な色の濃い転換社債(CB)，新株引受権付社債(WB)など，多くのバリエーションが存在している。これに加えて，取り引きの方法にも一般的な現物のほかに，先物，オプションなどがあるからこれはもう，複雑怪奇そのもの。

それは認識したうえで，ここではあえて株式だけに焦点を絞ってみようと思う。前にもいったように，細かく，正確なのだけがいいとはいえないのだ。本当に必要なポイントに絞って，初めて見えてくるものもあるのだから。

皆さんは株式についてどのぐらいご存じだろうか。NTT株やブラックマンデーぐらいなら知っているって。上出来です。なになに，JR株の上場が延期されそうだって。かなりお詳しいですね。

何年前かまでは，株なんてなにか怪しげで，ごく一部の人だけがやっているものだった。ところが時代は変わるもので，いまや普通のおじさん，おばさんから始まって，OL，はてはアルバイト成金の学生までが株

に手を染める時代だ。

これにはもちろん，時代背景というものもある。株価の上昇はとどまるところを知らず，手っとりばやく，誰でも儲けることができる時代だった。NTT株が一般の人をひきよせた，ということも忘れてはならない事実だ（いまでこそ見る影もないが）。

ところが，皆さんご存じのように，「ブラック・マンデー（暗黒の月曜日）」なる株価の大暴落が起こった。さらに追い打ちをかけるべく，昨年初頭にも再び大暴落が起こった。こうして「確実に儲かる」という神話は消え，空前の活気を呈していた証券取引所もいまでは閑散としている。

結局現在に至るまで，株価はこれといった回復の兆しを見せていない。しかし，株が一般的なものになったという事実に変わりはない。フツーのOLが結婚資金を貯めるべく，馬券感覚で株を買う時代なのだ。

そんな時代なのだから，というわけでもないのだが，きわめて個人的な関心から，最近僕は株式について，もっと広くいえば有価証券そのものについて勉強している。

ご時勢がら，株には株価という「時価」があって，皆その上がり下がりに一喜一憂している，ということぐらいはご存じだろう。株価というものをじかに見てみたいのであれば，新聞を開いてみるといいだろう。真ん中あたりに，株価欄があるはずだ。

なになに見つからないって，残念，その新聞はハズレだ。というのは冗談だが，スポーツ新聞には載っていないし，たとえ天下の日経新聞でも，土曜の夕刊，日曜，月曜の朝刊には載っていないはずだからご注意を（理由は自分で考えてみたまえ）。

さて，株価が株の「時価」であることはいいとしよう。それにしても，何を基準に時価が決まっているのだろう。それに，株式とはいったいなにものなんだろう。

stock

今回はこんな疑問から，シミュレーションへの道が始まる。

とにもかくにも，「株」とはどんなものなのかを考えてみることから始まる。証券取引法第2条によれば，株券とは国債，地方債，社債，投資信託と貸付信託の受益証券と並んで，有価証券のひとつということになっている。

といわれても，全然ピンとこないのが普通だろう。もっと一般的にいうと，株とは企業の発行している券，ということになる。券といってもただものではない。株式会社の資本の源なのだ。

株式会社，というのはよく聞く言葉だろう。皆さんがご存じの会社の大部分はいわゆるこの「株式会社」だ。

ある人が，会社を作ろうと思い立ったとする。しかし，会社を作るためには，やはりそれなりに先立つものが必要になってくる。つまり，お金＝資本だ。

大きい会社を作るためには，当然それだけ多くの資本を必要とする。ひとりではとても賄い切れない金額だ。そこで，出資者が必要になってくる。

その新会社にお金を提供（出資）する人。こういった人が何人か集まって，初めて「株式会社」を設立することができる。まずはめでたし。しかし，会社のほうとしても，お金を出してもらった以上，ありがとうだけではすまされないことになる。

事業活動を行い，利益を上げ，出資者に還元しなければならないのだ。一方，出資者としてもお金を出すだけではすまない。出資した会社が，万が一，潰れてしまったら，お金は返ってこないのだから。

つまり，両者ともに，それぞれメリットとリスクを抱えることになるのだ。この一見不思議な，持ちつ持たれつの関係を仲介しているのが，噂の「株（券）」というものだ。出資者は出資したお金の代わりとして，株券を受け取る。この結果，株を所有する者，すなわち「株主」になる。一方，会社は株と引き替えで集めた資本で，事業を始めることができる。株で成り立っている会社だから，株式会社，を名乗ることになる。こうして，出資を受けた「株式会社」と，出資した「株主」という関係が「株」を通して成立する。

ここでもう一度，「株式会社」と「株主」それぞれの権利と義務について振り返ってみることにする。

まず，株式会社は，出資された「資本」を使う権利を得るが，それと同時にその「資本」を元に事業を展開し，利潤を追求する義務を負う。さらに得た利潤を株主に還元しなければならない。

一方の株主としては，企業が得た利潤を分配してもらう権利を得る。その代わりに，会社が経営に行き詰まって倒産した際には，出資したお金を放棄しなければならない。

value

さて，株式会社と，株主を媒介するものとしての株の意味はわかった。それではその株になぜ値段がつくのだろうか。株とは，いったいどんな価値を持つのだろうか。

株主は株を買うことによって，株式会社に対する出資を行う。会社にとってみれば，返す必要のないお金である。もちろん，配当という形で，利潤の還元も行う。

しかし，昨今の株価水準からすれば，配当だけを目当てに出資するのは無駄が多い。配当とはいっても，天下のトヨタでもたかだか1株当たり10円程度である。ところが，現在トヨタの株価は約1,500円だ。1,500円投資して，半年に10円では年間での利回りは2%にもならない。

これならば，よっぽど銀行にお金を預けておいたほうが得になってしまう。しかし，株に投資した場合，以下のメリットが見込めるといわれる。すなわち，

・配当（利潤の還元）

出資に対する見返りとして，会社側が行う。インカム・ゲインとも呼ばれ，比較的コンスタントに見込める収入ではあるが，現在の株価水準からすると，妙味は薄い。

・株価の上昇による差益

現在の株式投資の主目的ともいえるのが，この差益。インカム・ゲインに対して，キャピタル・ゲインと呼ばれる。株価の変動によって，大きな利益を産むこともあれば，逆に莫大な損失（キャピタル・ロス）を産むこともある。

株が「バクチ（もっと洗練されたいい方でいえば『リッチマンズ・ギャンブル』）」と呼ばれるゆえんである。

具体的な例で見てみよう。有名なNTT株。最初は1987年2月に1株1,197,000円で売りに出された。これが同年4月には3,180,000円を記録。ということは，1,200,000円で購入して，最高値3,180,000円で売った人は，なんと2，3カ月で2,000,000円近くのキャピタル・ゲインを得たことになる。年利でいえば700%以上だ。

ところが，現在，そのNTT株も1株800,000円を切る有様。仮に最高値3,180,000円で購入し，いままだ保有しているとすると，実に2,400,000円近くの損ということになる（NTT株を100株近く持っていて，大損をしたプロ野球選手の噂もあるし……）。

この落差の大きさこそが，株式投資の醍醐味であるし，同時に恐ろしさでもある。

game

少し，話が長くなりすぎたかもしれない。そろそろ，ここらで，本題のシミュレーションに入っていくことにしよう。

物事を学ぶ場合，手っ取り早い方法はと

いえば，やはり実際に挑戦してみることだと思う。しかし，ことが株式投資となると，最低でも1,000,000円ぐらいの元手が必要だし，なによりもリスクが大きすぎる。

さすがに，「さあ，みんなやってみましょう」とお勧めするわけにもいかない（証券会社の回し者じゃないんだし，第一責任は持てないぞ）。

そこで登場するのが，例によって例のごとく，架空の世界を作り上げるシミュレーションというわけだ。

まあ，まずは今月のプログラムを入力してほしい。内容のわりには長いような気もするが，まあ，まだ夏休みという方も多いであろうから，気長にやってください。

使い方はそれほど難しくはないはず。基本的にすべてメニュー方式だから，メニューにしたがっていけば，ひと通りの使い方はわかるだろう。

まず，口座を開設する。これは銀行と同じだ。その際には，手持ち資金を決めなければならない。少なすぎれば株を買うことができないし，多すぎてもスリルがなくてつまらないだろう。

2，3千万円が妥当なところかな。さて，これを元手に株を売買し，できるだけ多くの利益を上げることが目標だ。まあ，とりあえずは買ってみよう。

何を買うか。新聞の株式欄をじっくり見て，まずは，何でも好きな会社を選べばいいだろう。ここでは敬意を表して，シャープといってみましょうか。

なになに，シャープがどこにあるかわからないって。「電気機器(あるいは電機，新聞によって表現は多少異なる)」のところをじっくりと探してごらん。必ず見つかるはず。

見つかっても，横に並んだ数字にとまどう方もいらっしゃるだろう。ひと口に株価といっても（朝刊であれば），4種類の数字が載っているはずだ。

左から，始値，高値，安値，終値の順だ。朝刊の場合，載っている株価は前日1日で取り引きされた株価。始値は前日取り引きが始まったときについた値段。終値は逆に取り引きが終了したときの値段だ。高値，安値は想像がつくだろうが，1日のうちでついた，最も高い値と安い値のこと。

これらどれもが株価には違いないから，どれを選んでもいいわけではあるが，ただひとつ注意しなければならないのは，基準を統一しなければならないということ。

買うときは安値で，売るときは高値で，としてしまうと，たしかに利益は膨らむものの，あまり現実を反映していないことになる。思ったとおりのタイミングで売り買いできるほど，甘いものではないのだ。

考えてみればわかるように，すでに前日のことになってしまったからこそ，これが高値，これが安値といえるのであって，現実にその場にいたのであれば，いつが高値でいつが安値，なんてわかるはずがない。

というわけで，ここは，基準をただひとつ「終値」に固定することにしよう。以降は，買うときも売るときも終値をその値段とすることにしよう。

buy

さて，それではシャープを買うことにしよう。7月23日付け日本経済新聞朝刊によれば（とはいっても新聞によって株価が違うことなどありえないが），7月22日の取り引きでのシャープの終値は1,550円とのこと。よし，株価は1,550円と。

次なる問題は何株買うか，ということ。株はたしかに1株単位で存在するし，株価も1株の値段ではあるのだけれど，だからといって，1株単位で買うことができるか，というとそうでもない。

株の売り買いにおいては，売買単位というものが存在し，東京証券取引所（いわゆる東証）などの取引所を介する場合には，この売買単位に従ってでないと，売り買いできないことになっているのだ。

ほとんどの株の売買単位は1,000株。つまり，1,000株，2,000株といった単位でないと売買ができないということになる。もっとも，1,000株単位ということになると，お値段もそれなりに張ってくる。

まあ，平均的には7～80万円から120～130万円といったところだろうか。これが個人が株式投資を敬遠する一因とはなっている。そこで，最近では個人でも買いやすいように売買単位を100株にするなどという動きも見られる，……というのはまったくの余談。

さらに余談をすると，売買単位が1,000株でないいちばん有名な例が，前にも出てきたNTT株で，これは売買単位が1株。当然といえば当然で，安くなった現在でも1株80万円前後だもの，1,000株単位でしか買えないとなったら，最低でも8億円（！）必要になってしまうものね。

さてさて，ここではど～んと10,000株買ってみることにしましょうか。こういうところで豪快にできるのが，シミュレーションのいいところだよね。

もし本当に自分のお金がかかっていたら，こうはいかない。

買ったはいいのだが，支払うお金は，株価×株数でいいのだろうか。残念ながらそう簡単にはいかない。手数料だの，税金だのが必要になってくるのですよ。

表1に，株式売買に関する項目をまとめておいた。今回は“買い”であるから，約定金額，委託手数料，消費税の3つの項目が絡んでくることになる。

実際の売買では証券会社が計算してくれるし，今回のシミュレーションでも当然すべて計算されて出てくるが，ここはひとつ，後学のためにも自分で計算してみることにしよう。

まず，約定金額。これは簡単で，株価×株数だから，今回の場合

1,550 × 10,000

で，1550万円ということになる。

委託手数料は，表2を利用して，

1550万円×0.575％＋25,000円

すなわち，114,125円ということになる。

消費税は委託手数料の3％であるから，3,423円となる（円未満切り捨て）。

結局，シャープの株を10,000株買うためには，

1550万円＋114,125円＋3,423円

という具合で，15,617,548円必要なことがわかる。

ずいぶん手間が(それにもちろんお金も)かかったが，これでシャープ株10,000株が

表1　株式売買の基礎項目　（円未満切り捨て）

約定金額	＝（取引が成立した）株価×株数
委託手数料	：算出表（表2）参照
消費税	＝委託手数料×3％
有価証券取引税	＝約定金額×0.3％
みなし売却益	＝約定金額×5％
売却益税	＝みなし売却益×20％
買い：買い代金	＝約定金額＋委託手数料＋消費税
売り：売り代金	＝約定金額－委託手数料－消費税－有価証券取引税－売却益税

表2　株式委託手数料簡便算出表

1990年6月4日約定以降適用

約定金額	手数料率(％)	加算額(円)
～100万円以下	1.150	0
100万円超～500万円以下	0.900	＋2,500
500万円超～1,000万円以下	0.700	＋12,500
1,000万円超～3,000万円以下	0.575	＋25,000
3,000万円超～5,000万円以下	0.375	＋85,000
5,000万円超～　1億円以下	0.225	＋160,000
1億円超～　3億円以下	0.200	＋185,000
3億円超～　5億円以下	0.125	＋410,000
5億円超～　10億円以下	0.100	＋535,000
10億円超～	0.075	＋785,000

1．約定金額の1.150％に相当する額が2,500円に満たない場合は2,500円。2．新株引受権証書，優先株及び新株引受権証券の委託手数料は上記と同じ。

君のものになったわけだ。

sell

さて翌日，こんな高価なものなど持ち慣れない君は，早くも売りたくてうずうずしているはず。それじゃあ，売りましょう。

翌日の終値は，と。今回は3通り考えてみることにしよう。

1) 残念，株価が下がった。終値は1,450円（前日比100円安）。

まあ，いちばん悲惨なパターンである。シャープの名誉のためにいっておくと，優良株シャープのことだから，こんなことはめったにおこらない「はず」だけどね。

ともかく売ってみようか。表1からわかるように，売る場合はさらにいろいろな項目がかかわってくる。

約定金額	1450万円
委託手数料	108,375円
消費税	3,251円
有価証券取引税	43,500円
（みなし売却益	725,000円）
売却益税	145,000円

結局，売って入ってくる金額は

約定金額－委託手数料－消費税－有価証券取引税－売却益税＝14,199,874円

ということになる。

ここで，10,000株買うために15,617,548円使っているのだから，なんと1日で1,417,674円の大損をしたことになる。さすが株式，恐ろしいものである。

2) もう1日我慢したところ，株価が1,550円に戻った。つまり，株価の変動は±0。1日我慢してよかった。と，ほっとするのはまだ早い。とりあえず，この時点で売るとどういうことになるか見てみよう。

約定金額	1550万円
委託手数料	114,125円
消費税	3,423円
有価証券取引税	46,500円
（みなし売却益	775,000円）
売却益税	155,000円

結局，売って入ってくる金額は

約定金額－委託手数料－消費税－有価証券取引税－売却益税＝15,180,952円

ということになる。

おやおや，これでも436,596円の損をしたことになる。え，納得いかない。たしかに，株価は±0だけれども，手数料だの，税金だのを持っていかれていることを忘れてはならない。

要するに，この損436,596円は，まるまる手数料と税金の分なのだ。つまり，証券会社と税務署は必ず儲かる，という仕組みになっているのだ。

3) 次の日，ついに株価は1,650円になった。つまり，買ったときに比べ，100円上がったことになる。

この場合はさてどうなりますか。

約定金額	1650万円
委託手数料	119,875円
消費税	3,596円
有価証券取引税	49,500円
（みなし売却益	825,000円）
売却益税	165,000円

結局，売って入ってくる金額は

約定金額－委託手数料－消費税－有価証券取引税－売却益税＝16,162,029円

今回はめでたく，544,481円の利益を生じたことになる。

もうお気づきだとは思うが，株の売買では，投資家は手数料だの税金だののおかげで，損をしやすい仕組みになっている。そのかわり，儲かったときのリターンも大きい。ようするにハイリスク・ハイリターン商品（危険は大きいが，儲かったときは利益も大きい）の典型といえる。

参考までに少しつけ加えておくと，損をしたときにまで「売却益税」，すなわち利益に対する税金がかかるのはおかしいじゃないか，と思われた方も多いだろう。

これは「みなし売却益」という仕組みに原因がある。本来は利益に対しての税金であるから，株式の取引のたびに，正確に利益を算出し，税金も計算するべきである。しかし現実問題として，これは非常に手間がかかる。そこで，株取引があった場合，約定金額の一律20%を儲け分としましょう。そして税金もその「みなし」儲け分についてのみ課税しましょう，ということで生まれたのが「みなし売却益」制度である。税務署にとってみれば，証券会社が代行して税金を徴収してくれるのであるから，手間はかからないし，確実に課税することができる。

投資家としても，大きな儲けを出した場合にも（たとえば，株価が2倍に跳ね上がり，約定金額と同じだけの利益を出したとしても），20%分だけに対する課税ですむ，さらに自分の名前が表に出ない（あまり公明正大なメリットではないが）というメリットがある。

逆に，最初の例のように損をした場合には，この「みなし売却益制度」は踏んだり蹴ったりの制度といえる。そこで，もちろん自分で税務署に申告する「申告課税」という制度も存在しており，これは投資家が自由に選択できることになっている。

ups and downs

余談が長くなってしまったが，こうやって実際に（シミュレーションではあるが）株の売買を体験してみると，実にスリリングなゲームであることがわかるだろう。

しかも，この「ゲーム」の面白いところは，株価の変動がある要因に基づいている，というところにある。

よくあるゲームのように，乱数やサイコロで決まっているわけではないのだ。

それではどんな要因があって，それによってどのような変動をするのか，ということになってくると，これはまったくの自由研究の範囲になってくる（ちょうど夏休みだし。しかし，夏休みの自由研究で「株」について研究を行っていたら，これは驚くだろうなあ）。

まあ少しだけ有名な例を挙げておこうか。たとえば，為替レート。円高とか円安とかのあれだ。円高になった場合。これは石油などを安く買えることになるから，電力会社が儲かるだろう，という読みが働き，電力会社（東京電力など）の株価が上がったりする。

逆にホンダなどは輸出が多いから，円高によって利益が圧縮されるだろうという読みで，株価が下落したりする。

ほかにも金利の影響。金利が上がると，借金の多い会社は経営が苦しくなる（返さなければならない利子が増えてしまう）。これを反映して株価が下がる。一方，無借金経営の超優良会社トヨタなどは，逆に預金につく利子が増えるから儲かることになる。当然，株価は上がる。

もっとも，いまや，株価にはこんな単純な理論は通用しない。何百，何千という多くの要因が複雑に絡みあって，初めて株価が形成されている。

だから，同じように石油の値段が変わっても，火力発電への依存度によって，株価の変動具合が変わってくる，なんていうのは朝飯前である。

だから，一見まったく関係ないような要因によって，突然ある銘柄の株価が変動するという，まさしく「風が吹けば桶屋が儲かる」的な話は，証券界にはゴロゴロしている。

経済的な動向，政治的な動向，市場の内部動向，さらには根も葉もない噂など，株価は非常に多くの要因によって左右されている。

それだけに，読むのは難しいし，いまだかつて，株価を完全に読み切った人など存在しない。もし読むことができれば，確実に儲かり，巨額の富を手にすることができる。これが多くの人を株式市場にひきつけてきた理由である。

いってしまえば，ギャンブルということに変わりはない。しかも，非常に高度で，非常に知的な。なかには，ニューロ・コンピュータを利用して株価予測をしようという動きもある。

さらには，ニューロの次に来るモノ，「カオス」理論まで持ち出そう，という話も耳にする。

step up

非常に知的なゲームだと思う。読みどおりに相場が動いたときのリターンは莫大なものがある。逆に，読みを外したときのリスクが大きいのも，また事実だ。

しかし，公正で，誰にでも参加できる市場を目指しての改革も進んでいる。「ローリスク・ハイリターン」商品である，オプション商品の導入などがいい例である。

もっとも，一連の不祥事に見られるように，依然として，なにやらあやしげで，ドロドロした側面を持つのもたしかだろう。

結局は，自分でどこまで責任を取れるか，ということだ。証券会社のお勧めに従うのもいいだろう。ただし，それを最終的に決断するのは本人である。

自分の判断を信じ，その判断に対し，責任を取れるようになって初めて，このリスキーなゲームに参加する資格を得るのだ。それまでは，せめてこのシミュレーションでじっくりと研究を積んでほしい。そして，本当にやりたければ，いつか本物の株式取引に挑戦してみるといいだろう。

ただし，僕は「一切」責任は取りませんから，そのつもりで。せいぜい「ハイリスク・ノータリン」にならないように気をつけてください。ちなみに僕はまだ2，3年，これで勉強を続けるつもりです。

thanks to……

いま，こうして6回におよぶ連載をなんとか終えることができて，正直なところホッとしている。

僕にとっては，連載というのは初めての経験だったが，面白く，やりがいのあるものであると同時に，非常にプレッシャーが大きくつらいものでもあった。

だいぶブランクも長く，回り道をしてしまったが，とにもかくにも，いちおうの終止符を打つことができた。

これも，拙著に温かい声援を送ってくださった読者の皆さんの「力」だと心から感謝している。そして，筆者の我儘をきいてくださった，編集部の皆さん。僕だったら，とっくに堪忍袋の緒が切れているはずだ。

さきほど，終止符と書いたが，決してすべての終わりというわけではない。終わりは次なるステップに向かってのスタートにほかならないのだから。

僕はこれからも，多くの新しいことを学んでいくつもりだ。経済，政治，物理，スポーツ，語学，何でも結構。

その過程で僕が学びとったことを，またこのような場で皆さんにお伝えできればと思っている。

これは皆さんにもいえるはずだと，僕は信じている。最近，何か新しいことを始めましたか。現状に満足しきっていませんか。次なるステップへ，足を踏み出してみてはいかがですか。誰が何といおうと，いまは夏，飛躍のときなのだから。

リスト1

```
1000 /* Investment Simulation
1010 /*
1020 /*             Stock Investment
1030 /*
1040 /*                       1991.7. (C) Cammon
1050 /*
1060 /* intialize
1070 width 96
1080 console 0,32,0
1090 int i,fno
1100 char menuno,customno
1110 char endflag
1120 dim str customname(10)
1130 dim int amount(10)
1140 dim int passwd(10)
1150 dim int stockno(10)
1160 dim str stockname(10,20)[20]
1170 dim str stockdate(10,20)[10]
1180 dim int stockam(10,20)
1190 dim int stockpr(10,20)
1200 dim int stockut(10,20)
1210 dim int stam(20),stpr(20),stut(20)
1220 dim str menu(5)
1230 menu(1)="取引"
1240 menu(2)="口座状況照会"
1250 menu(3)="新規口座開設"
1260 menu(4)="口座廃止"
1270 menu(5)="終了"
1280 str fname = "a:stock.dat"
1290 /* load data
1300 error off
1310 locate 2,1: print "19";date$;" (";day$;")"
1320 locate 65,0: print "株式投資シミュレーション"
1330 console 2,30,0
1340 fno = fopen(fname,"r")
1350 if fno <> -1 then dload(fno)
1360 /* menu
1370 repeat
1380   cls
1390   locate 40,5: print "総合メニュー"
1400   for i=1 to 5
1410     locate 35,7+2*i: print i;": ";menu(i)
1420   next
1430   locate 45,28: input "メニューを選択して下さい。 ",menuno
1440   switch menuno
1450     case 1 : trade()   : break
1460     case 2 : inquire(): break
1470     case 3 : open()    : break
1480     case 4 : close()   : break
1490     case 5 : quit()
1500   endswitch
1510 until endflag
1520 /* save data
1530 fno = fopen(fname,"c")
1540·dsave(fno)
1550 console 0,31,1: cls
1560 end
1570 /* menu 1 : 取引
1580 func trade()
1590   str i_com
1600   cls: locate 40,3: print "取引"
1610   select = check()
1620   if select then show(select) else return()
1630   repeat
1640     erase(45,28): i_com = ""
1650     input "売り(S) / 買い(B)  ",i_com
1660     if instr(1,"Ssト",i_com) and stockno(select)>0 then sel
l(select)
1670     if instr(1,"Bbコ",i_com) then buy(select)
1680   until i_com = ""
1690 endfunc
1700 /* menu 2 : 口座状況照会
1710 func inquire()
1720   cls: locate 40,3: print "口座状況照会"
1730   select = check()
1740   if select then show(select): confirm("")
1750 endfunc
1760 /* menu 3 : 新規口座開設
1770 func open()
1780   str i_customname
1790   int i_amount,i_passwd,i
1800   repeat
1810     cls: locate 40, 8: print "新規口座開設"
1820     if customno = 10 then ( confirm("口座が一杯です。")
1830                             return() )
1840     repeat
1850       i_customname = "": erase(35,12)
1860       input "お名前   : ",i_customname
1870     until i_customname <> ""
1880     repeat
1890       i_amount = 0: erase(35,14)
1900       input "入金額   : ",i_amount
1910     until i_amount >= 1000000
1920     repeat
1930       i_passwd = 0: erase(35,16)
1940       input "暗証番号 : ",i_passwd
1950     until i_passwd > 0
1960   until ask("上記口座を開設します。") = 0
1970   for i = 1 to 10
1980     if customname(i)="" then (
1990         customname(i) = i_customname
2000         amount(i) = i_amount
2010         passwd(i) = i_passwd
2020·        customno = customno + 1
2030         break )
2040   next
2050   cls: show(i): confirm("")
2060 endfunc
2070 /* menu 4 : 口座廃止
2080 func close()
2090   str pamount
2100   cls: locate 40,3: print "口座廃止"
2110   select = check()
2120   if select then show(select) else return()
2130   if stockno(select) > 0 then (
2140       confirm("株式を売却して下さい。"): return() )
2150   if ask("上記口座を廃止します。") then return()
2160   pamount = yen(amount(select))
2170   confirm("残高は "+pamount+"でした。")
2180   customname(select) = "": stockno(select) = 0
2190 endfunc
2200 /* menu 5 : 終了
```

```
2210 func quit()
2220   endflag = 1
2230 endfunc
2240 /* 口座リスト
2250 func char check()
2260   int i,j,i_passwd
2270   char select
2280   locate 40,5: print "口座リスト"
2290   if customno = 0 then {
2300             confirm("口座はありません。")
2310             return(0) }
2320   for i = 1 to 10
2330     if customname(i) <> "" then {
2340          j = j + 1: locate 35,5+2*j
2350          print digit2(i);" : ";customname(i) }
2360   next
2370   erase(55,28)
2380   input "口座番号を選択して下さい。 ",select
2390   if customname(select) = "" then return(0)
2400   erase(0,5)
2410   locate 30,10: print "口座番号 : ";digit2(select)
2420   locate 30,12: print "お名前   : ";customname(select)
2430   locate 30,14: input "暗証番号 : ",i_passwd
2440   if i_passwd = passwd(select) then return(select)
2450   return(0)
2460 endfunc
2470 /* 口座内容
2480 func show( select;char )
2490   int i
2500   erase(30,5): print "口座番号 : ";digit2(select)
2510   locate 30, 7: print "お名前   : ";customname(select)
2520   locate 30, 9: print "現金残高 : ";yen(amount(select))
2530   locate 30,11: print "保管株式 : ";digit2(stockno(select));
"件"
2540   for i = 1 to stockno(select)
2550     locate 5,12+i
2560     print digit2(i);" : ";stockname(select,i);" ... ";
2570     print digits(stockam(select,i),6);"株  ";
2580     print yens(stockpr(select,i),16);"( @";
2590     print yens(stockut(select,i),10);" )    ";stockdate(se
lect,i)
2600   next
2610 endfunc
2620 /* load data
2630 func int dload( fno;int )
2640   int i,j,counter
2650   str i_customname,i_stockname,i_stockdate
2660   for i = 1 to 10
2670     freads(i_customname,fno): customname(i) = i_customname
2680     if i_customname <> "" then customno = customno + 1
2690   next
2700   fread(passwd,11,fno)
2710   fread(amount,11,fno)
2720   fread(stockno,11,fno)
2730   for i = 1 to 10
2740     for j = 1 to stockno(i)
2750       freads(i_stockname,fno): stockname(i,j)=i_stockname
2760       freads(i_stockdate,fno): stockdate(i,j)=i_stockdate
2770     next
2780     counter = stockno(i) + 1
2790     fread(stam,counter,fno)
2800     fread(stpr,counter,fno)
2810     fread(stut,counter,fno)
2820     for j = 1 to stockno(i)
2830       stockam(i,j) = stam(j)
2840       stockpr(i,j) = stpr(j)
2850       stockut(i,j) = stut(j)
2860     next
2870   next
2880   fcloseall()
2890 endfunc
2900 /* save data
2910 func int dsave( fno;int )
2920   int i,j,counter
2930   str o_customname
2940   for i = 1 to 10
2950     fwrites(customname(i),fno)
2960     fputc(&HD,fno): fputc(&HA,fno)
2970   next
2980   fwrite(passwd,11,fno)
2990   fwrite(amount,11,fno)
3000   fwrite(stockno,11,fno)
3010   for i = 1 to 10
3020     for j = 1 to stockno(i)
3030       fwrites(stockname(i,j),fno)
3040       fputc(&HD,fno): fputc(&HA,fno)
3050       fwrites(stockdate(i,j),fno)
3060       fputc(&HD,fno): fputc(&HA,fno)
3070       stam(j) = stockam(i,j)
3080       stpr(j) = stockpr(i,j)
3090       stut(j) = stockut(i,j)
3100     next
3110     counter = stockno(i) + 1
3120     fwrite(stam,counter,fno)
3130     fwrite(stpr,counter,fno)
3140     fwrite(stut,counter,fno)
3150   next
3160   fcloseall()
3170 endfunc
3180 /* buy
3190 func buy( select;int )
3200   int stno,i_stockam,i_stockut
3210   int price,consump,net
3220   str i_stockname
3230   erase(10,30): print "[ 買 ]"
3240   repeat: erase(20,30): input "銘柄名 : ",i_stockname
3250     until i_stockname <> ""
3260   repeat: erase(50,30): input "株数   : ",i_stockam
3270     until i_stockam > 0
3280   repeat: erase(70,30): input "単価   : ",i_stockut
3290     until i_stockut > 0
3300   price = i_stockam * i_stockut
3310   consump = comission(price) * 0.03#
3320   net = price + comission(price)+ consump
3330   erase(0,5)
3340   locate 30, 7: print "銘柄          : ";i_stockname
3350   locate 30, 9: print "株数          :";i_stockam;"株"
3360   locate 30,11: print "単価          : ";yen(i_stockut)
3370   locate 30,13: print "約定金額   : ";yen(price)
3380   locate 30,15: print "委託手数料 : ";yen(comission(price))
3390   locate 30,17: print "消費税       : ";yen(consump)
3400   locate 30,19: print "代金合計    : ";yen(net)
3410   if ask("買い付けを実行しますか") then show(select): return()
3420   if net > amount(select) then {
3430       confirm("資金が足りません。"): show(select):return() },
3440   stockno(select) = stockno(select) + 1
3450   stno = stockno(select)
3460   stockname(select,stno)=left$(i_stockname+space$(20),20)
3470   stockam(select,stno) = i_stockam
3480   stockut(select,stno) = i_stockut
3490   stockdate(select,stno) = date$
3500   stockpr(select,stno) = net
3510   amount(select) = amount(select) - net
3520   show(select): confirm("")
3530 endfunc
3540 /* sell
3550 func sell( select;int )
3560   int stno,i_stockam,i_stockut
3570   int net,price,consump,tradetax,captgain,gaintax
3580   str i_stockname
3590   erase(10,30): print "[ 売 ]"
3600   repeat: erase(20,30): input "銘柄番号 : ",stno
3610     until stno < 11 and stockname(select,stno) <> ""
3620   repeat: erase(50,30): input "株数   : ",i_stockam
3630     until i_stockam > 0 and i_stockam <= stockam(select,st
no)
3640   repeat: erase(70,30): input "単価   : ",i_stockut
3650     until i_stockut > 0
3660   price = i_stockam * i_stockut
3670   net = price - comission(price)
3680   consump = comission(price) * 0.03#
3690   captgain = price * 0.05#: gaintax = captgain * 0.2#
3700   tradetax = price * 0.003#
3710   net = net - consump - tradetax - gaintax
3720   erase(0,5)
3730   locate 30, 7: print "銘柄             : ";stockname(select,
stno)
3740   locate 30, 9: print "株数             :";i_stockam;"株"
3750   locate 30,11: print "単価             : ";yen(i_stockut)
3760   locate 30,13: print "約定金額         : ";yen(price)
3770   locate 30,15: print "委託手数料       : ";yen(comission(price
))
3780   locate 30,17: print "消費税           : ";yen(consump)
3790   locate 30,19: print "有価証券取引税 : ";yen(tradetax)
3800   locate 30,21: print "売却益税         : ";yen(gaintax)
3810   locate 30,23: print "代金合計         : ";yen(net)
3820   if ask("売却を実行しますか") then show(select): return()
3830   stockpr(select,stno) = 1# * stockpr(select,stno) / stock
am(select,stno) * (stockam(select,stno) - i_stockam)
3840   stockam(select,stno) = stockam(select,stno) - i_stockam
3850   amount(select) = amount(select) + net
3860   if stockam(select,stno) = 0 then dismiss(select)
3870   show(select): confirm("")
3880 endfunc
3890 /* dismiss
3900 func dismiss( select;int )
3910   int i,j
3920   stockno(select) = stockno(select) - 1
3930   for i = 1 to stockno(select)
3940     if stockam(select,i) = 0 then {
3950       for j = i to stockno(select)
3960         stockname(select,j)=stockname(select,j+1)
3970         stockdate(select,j)=stockdate(select,j+1)
3980         stockam(select,j)=stockam(select,j+1)
3990         stockpr(select,j)=stockpr(select,j+1)
4000         stockut(select,j)=stockut(select,j+1)
4010       next: break }
4020   next
4030   stockname(select,stockno(select)+1)=""
4040 endfunc
4050 /* 確認
4060 func char ask( message;str )
4070   str sure
4080   erase(45,28): print message;"  [ CR / N ] ";
4090   sure = inkey$
4100   if sure = chr$(&HD) then return(0) else return(1)
4110 endfunc
4120 func str confirm( message;str )
4130   str sure
4140   erase(45,28): print message;"  [ 確認 ] ";
4150   sure = inkey$
4160   return(sure)
4170 endfunc
4180 /* 数字表示
4190 func str digit2( i;int )
4200   str digit
4210   digit = right$("0"+str$(i),2)
4220   return(digit)
4230 endfunc
4240 /* 金額表示
4250 func str yen( i;int )
4260   str amount
4270   int oku,man,ichi
4280   oku = i / 100000000#
4290   man = ( i - oku * 100000000# ) / 10000#
4300   ichi = i - ( oku * 100000000# + man * 10000# )
4310   if oku then amount = str$(oku)+"億"
4320   if man then amount = amount + str$(man)+"万"
4330   if ichi then amount = amount + str$(ichi)
4340   if i then amount = amount + "円"
4350   return(amount)
4360 endfunc
4370 /* 委託手数料
4380 func int comission( pr;float )
4390   if pr <  217392     then return(2500)
4400   if pr <= 1000000    then return(pr*0.0115#)
4410   if pr <= 5000000    then return(pr*0.009#+2500)
4420   if pr <= 10000000   then return(pr*0.007#+12500)
4430   if pr <= 30000000   then return(pr*0.00575#+25000)
4440   if pr <= 50000000   then return(pr*0.00375#+85000)
4450   if pr <= 100000000  then return(pr*0.00225#+160000)
4460   if pr <= 300000000  then return(pr*0.002#+185000)
4470   if pr <= 500000000  then return(pr*0.00125#+410000)
4480   if pr <= 1000000000 then return(pr*0.001#+535000)
4490   return(pr*0.00075#+785000)
4500 endfunc
4510 /*
4520 func erase( x;char,y;char )
4530   locate x,y: print chr$(&H1A);
4540 endfunc
4550 func str digits( i;int,d;int )
4560   str digit
4570   digit = right$(space$(20)+str$(i),d)
4580   return(digit)
4590 endfunc
4600 func str yens( i;int,d;int )
4610   str amount
4620   amount = yen(i): amount = space$(20) + amount
4630   amount = right$(amount,d)
4640   return(amount)
4650 endfunc
```

Multiwordは救世主となるか

Ogikubo Kei 荻窪 圭

お待たせしました。やっとのことで，あの「マルチワープロ PRO-68K"Multiword"」を紹介できます。紹介するという予告を覆すことを何度繰り返したことか。さて，気になる出来のほうはどうでしょうか。

いやあ，面白かった。面白かった，というのともまた違うな。気持ちよかったというべきかもしれない。何がかって，荻窪圭がひさしぶりに観たロードショウ映画，「羊たちの沈黙」である。原作の面白さをここまで損なわないで作ってくれた監督と，ジョディ・フォスターと，アンソニー・ホプキンスと，スコット・グレンに感謝したいくらいだ。

そもそも，この映画は失われてしまったストイックさを十分に発揮している。まず，登場人物が自分で自分を語るシーンがない。感情的になって話をぶち壊すやつがいない。うっとうしい回想がない。不自然な男女のじゃれあいがない。観客の興味を煽るためだけの残虐なシーンや，カタルシスを得るためだけの格闘シーンもない。セックスアピールを意図したファッションの女が出てこない。ただただ，冷酷に冷静に話は進むのである。人間的と巷でいわれる感情は，すべて表情の奥に隠れている。簡単に感情を露出してしまうような甘いやつに，レクターの相手はできないのだ。

「傑作！」というほどの派手さはまったくないが(そこがまた凄いところだ)，あのクラリスとクロフォードとレクターがあれだけちゃんと描けていただけで私は許してしまう。本誌が出る頃にはおそらくロードショウは終わっているだろうな。残念なことに。

で，とにかく，世間に流通するほとんどのストーリーに足りないのがこういった冷酷とも思える表現，視点である。観客におもねった強引な演出はいらないのである。

私が"Multiword"に対して懸念しているのは，まさにその点だ（おお，なんと強引な展開)。

＊　＊　＊

本連載も今月で12回目だそうだが，ここにきてやっと「ワープロ」の話になるってのも，X68000の置かれている状況を物語っていて面白い。

そういうわけで，デバッグにあと10年はかかるのではないかと思われていた「マルチワープロ PRO-68K "Multiword"」が，ようやく完成しそうだ。私の手元には，その，ようやく完成しそうな"Multiword"がある。懸念したとおりである。しかし，世の中には観客におもねった感動の物語が好きな人が大勢いるように，人々は千差万別な十人十色である。

最初にいっておくが，X68000という小さな小さなマーケットで，この値段で，これだけ機能の詰まったワープロソフトが出るということは凄いことだ。機能を考えると，コストパフォーマンスは異常に高い。印刷することを考えると，ベストなパソコンはMacintoshだと私は思っているのだが，Macintoshのまっとうなポストスクリプト DTPソフトは10万円以上するし，私が愛用している図形作成機能もあって，マクロも書けて，という超多機能ワープロ "Solo Writer" は68,000円だ。さらに印字品質を求めようとすると，それなりのプリンタが必要で，印字速度を求めるとレーザープリンタは必須だ(Macintoshの場合，本体のほうでビットマップデータを作ってプリンタに流すから)。

普通のワープロソフトは（Macintoshの場合，普通の日本語ワープロでもマルチフォントでかなりWYSIWYGなのだが）5万円以上はする。X68000の10倍以上のマーケット規模を持つPC-9801でもこれだけの機能を持つソフトとなると，そうはない。

だが，問題は別のところにあったのだ。

1."Multiword"の3つの顔

もうご存じの人も多いと思うが，"Multiword"は3つのモードを持っている。ひとつがテキスト画面モード，もうひとつがマルチウィンドウ画面モード，そして，グラフィックエディタモードである。

グラフィックエディタというのはマジックパレット的なやつで，文書に貼り込むための絵を描くところだ。こいつだけは別プログラムになっている。

ということは，テキストとウィンドウ画面モードは同じプログラムというわけで，恐ろしいことにプログラムサイズは600Kバイトを超えるのだ。くわばらである。

"Multiword" とWP.Xの関係

当たり前の話だが，"Multiword" は多分にWP.Xを意識している。意識するのはいいのだが，WP.Xの，私はバグだと信じているうっとうしい禁則処理を見事再現してくれたのだ。禁則処理はする。するにはする。が，たったひとつしか禁則してくれないのだ。たとえば，行末に"。"がはみでたときにちゃんとやってくれ，とは私はいわない。こういうのは二重禁則という技だ。だが，"。改行"となったときに改行コードだけが次の行の先頭に来てしまうのは，言語道断である。改行コードは文字にあって文字にあらず。ここで楽をしてはいけないのだ。うるうる。二重禁則と "。+改行" は別なのだ。

さらに，ReturnとSHIFT＋Returnの動作が逆になってしまっている（改善か改悪か）ことを除けば，WP.Xで「んがー！」となった妙なカーソル移動は，そのまま互換性を持って継承されていると思っていい。TABキーはTABコードの挿入ではなく，単なるTAB位置へのジャンプである，とか，だ。

それだけではない。センタリングや右揃えといった行の整形機能も，WP.Xと同様，「スペースを適当に入れてセンタリングや右詰めに見せているだけ」である。行にセンタリングの属性がつくわけではない，というのもつけ加えておこう。

さて，テキスト画面モードというのは，一太郎というより，松というかVJE-Penといった趣だ。趣味の悪いことに，黒字に白で，文章の書かれていないところには黄色の中黒で埋めつくされ，ルーラーやページの区切りは水色という，“PC-9801のアプリケーションが好んで使う色合い”に染まっている。あな恐ろしや。

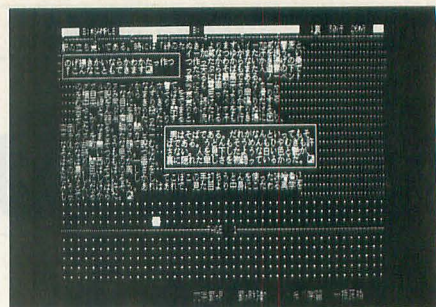
ポイントはウィンドウ画面だ。マルチウィンドウでバコバコとウィンドウが開き，WYSIWYG（だそうだ）で，グラフィックも挿入できてしまうという優秀さ。ああ，さすがである。

2.テキスト画面モード

まずはウィンドウ画面モードというのが“Multiword”を紹介するときのポイントなのだろうが，私の考えからいくと，やはり，ワープロの基本は文書入力であり，文書編集である。この基本を押さえないと，どんな機能の多いワープロでもただうっとうしいだけである。“Multiword”でもそれはわかっているのか，テキストモードとウィンドウモードの2つを持っている（ご丁寧にも，ウィンドウ画面モードではちゃんとグラフィック画面に文字を書いている）。だから，まずは基本から，ということだ。

いま，まさにそのテキスト画面モードで原稿を書いているのだが，どうも調子がでない。画面の構成が妙にPC-9801しているからか，とも思ったが，それだけでもないらしい。しばらくして原因がわかった。カーソルが点滅していないのだ。だから気持ち悪いのだ。とっても些細なことだが，点滅しないカーソルはやめてもらいたい，と，個人的にそう思う。

ぼやいてばかりも能がないので，本題に移る。

画面は写真1を見てもらうとして，操作体系だ。基本的に，“コマンドを先に選んでから操作対象を選ぶ”という日本の標準的な体系である。なおかつ，コマンドはメニューから選ぶのであるが，そのメニューはESCキーで画面最下段に現れるという日本伝統の一太郎式である。テキスト画面モードでは，高速化の見返りとして，マウスが使えない。ここが大きなポイントだ。たしかにスクロールはひゅんひゅんである（ただし，ひゅんひゅんなのはスクロールだけでROLL UP/DOWNによる書き換えや削除/挿入はさほどでもない）。

❶テキストモードの画面

ESCキーを押すと，画面最下段にメニューが出る。ここでカーソルキーで選ぶか，数字を押すと，さらに下層のメニューへ行く。という具合に，機能が多いから，どんどん下層へ下っていく。私がよく引き合いに出す文字列の削除の場合，

“ESCキーを押す→編集を選ぶ→文字列操作を選ぶ→カットを選ぶ→文字単位を選ぶ→始点を選ぶ→終点を選ぶ→Return”

という，なんともはやの8ステップが必要なのだ。これは一太郎と比べて2ステップも多い。体系としてはわかりやすいのだが，削除という頻繁に行う操作にしては，階層が深すぎる。さらに，途中で「あ，間違えた」というときには，また最初からやり直さねばならないのだ。機能が多いから，しょうがないといえばしょうがないのだがね。

ちなみに，メニュー体系を記しておいた（図1）。どれだけ機能が多いかは，これを見ればわかるようになっている。

図1　メインメニュー

Xメニュー
- A　電卓
- B　カラー設定
- C　メモリ残量
- D　カーソル速度
- E　キーリピート
- F　環境ウィンドウ

- A(1)編集
- B(2)文字設定
- C(3)罫線
- D(4)印刷
- E(5)ファイル
- F(6)その他

A(1)編集
- A(1)文字列操作
- B(2)カーソル移動
- C(3)行ぞろえ
- D(4)文字飾り
- E(5)ノーマル
- F(6)再変換

A(1)文字列操作
- A(1)カット
- B(2)コピー
- C(3)貼付(カット)
- D(4)貼付(コピー)
- E(5)検索
- F(6)置き換え

B(2)カーソル移動
- A(1)文の先頭行へ
- B(2)文の最終行へ
- C(3)任意の位置へ
- D(4)マーク位置へ
- E(5)マーク登録
- F(1)マーク消去

C(3)行ぞろえ
- A(1)改行幅変更
- B(2)文書幅変更
- C(3)右寄せ
- D(4)左寄せ
- E(5)センタリング

D(4)文字飾り
- A(1)1/4角上つき
- B(2)1/4角下つき
- C(3)1/4角上下
- D(4)横倍角
- E(5)縦倍角
- F(6)4倍角
- G(1)任意倍角
- H(2)文字色変更
- I(3)回転
- J(4)ライン
- K(5)かすれ
- L(5)網かけ
- M(1)強調
- N(2)リバース
- O(3)拡張字体

B(2)文字設定
- A(1)ノーマル
- B(2)大きさ設定
- C(3)装飾設定
- D(4)文字色設定
- E(5)回転角設定
- F(1)拡張字体設定
- G　設定ウィンドウ

B(2)大きさ設定
- A(1)1/4角上つき
- B(2)1/4角下つき
- C(3)横倍角
- D(4)縦倍角
- E(5)4倍角
- F(5)任意倍角

C(3)装飾設定
- A(1)ライン
- B(2)かすれ
- C(3)網かけ
- D(4)強調
- E(5)リバース

C(3)罫線
- A(1)外枠罫線
- B(2)罫線カット
- C(3)罫線コピー
- D(4)罫線貼付(カット)
- E(1)罫線貼付(コピー)
- F(2)線種変更
- G(3)通過罫線
- H(4)枠内操作
- I(1)移動状態
- J(2)罫線モード
- K(3)罫線色変更
- L(4)罫線領域解除

D(4)印刷
- A(1)印刷
- B(2)印刷条件設定
- C(3)書式設定
- D(4)印刷環境設定
- E　レイアウト表示
- F(5)縮小印刷

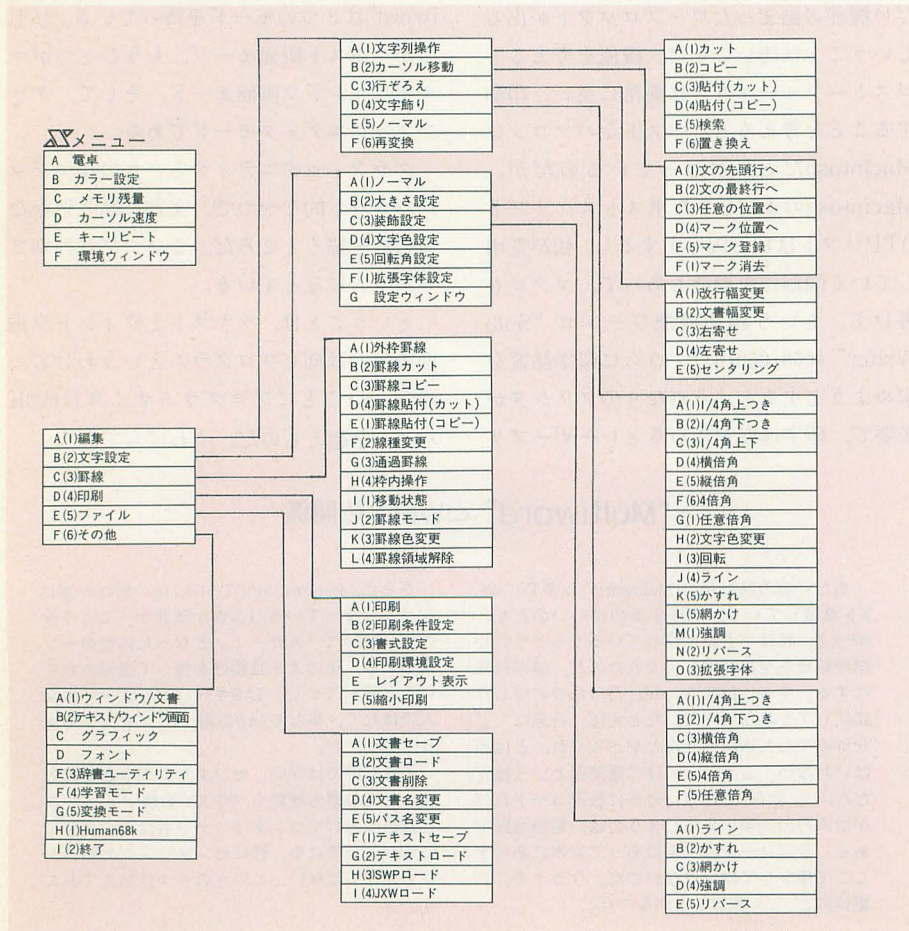
E(5)ファイル
- A(1)文書セーブ
- B(2)文書ロード
- C(3)文書削除
- D(4)文書名変更
- E(5)パス名変更
- F(1)テキストセーブ
- G(2)テキストロード
- H(3)SWPロード
- I(4)JXWロード

F(6)その他
- A(1)ウィンドウ/文書
- B(2)テキスト/ウィンドウ画面
- C　グラフィック
- D　フォント
- E(3)辞書ユーティリティ
- F(4)学習モード
- G(5)変換モード
- H(1)Human68k
- I(2)終了

3.ウィンドウ画面モード

ウィンドウ画面モードは，白地に黒の艶やかなマルチウィンドウだ。テキスト画面とは違ってマウスも使えるし，いくつもの文書を同時に開いて編集もできる。“Multiword”を立ち上げると，まずこの画面になる（写真2）。

ウィンドウ画面モードは，“Multiword”の売りを一身に背負っている。

たとえば，無意味なほど豊富な文字修飾機能だ。このあたりは写真3を見てほしいのだが，倍角やら4倍角はどうでもいい。

かすれやら網掛けやらの豊富さだ。任意倍角＋スムージングもやってくれる。しかし，そこまでやるならZ'sSTAFF PRO-68K ver.2.0のアウトラインフォントをサポートしてほしかったと心から思うところだ。CANVAS PRO-68KだってNEW Print Shop PRO-68Kだってサポートしているのだから，WYSIWYGなんていっている多機能ワープロは真っ先にサポートすべきだろう。

装飾とくれば，罫線である。罫線。日本のワープロは罫線に命をかけている。罫線の種類こそ７種類と多くはないが（十分多い気もするが），図１のコマンド表をみればわかるとおり，たくさんの罫線コマンドがある。使い分けははなはだむずかしい。

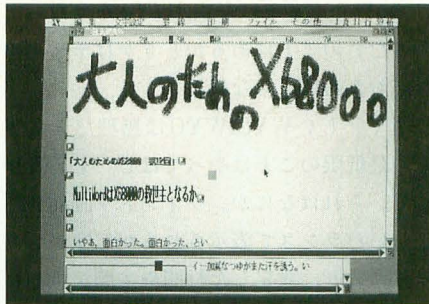
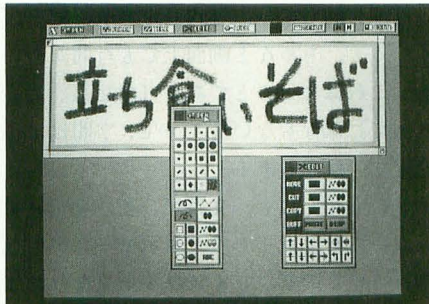
さて，そろそろ本題である。ウィンドウ画面での不思議な操作体系である。

テキスト画面では先にコマンドを指定して，最後にそのコマンドが処理する範囲を指定するというポストセレクション方式であったが，さすがにウィンドウ画面モードでは，そうもいかない。先に範囲指定するプリセレクション（日本語でいうと，範囲先指定）を無視できない。マウスを使ってのポストセレクションはいかにも間抜け。

“Multiword”ではなんと，両方ともサポートしている。まるで一太郎（一太郎が，実はプリセレクションもサポートしているというのは，あまり知られていない。もっとも，使いにくいので知っていても意味はないが）だ。

まず，なんの気なしに，ファンクションキーを押したり，マウスでメニューバーをクリックすると，ドロップダウンメニューが現れる。もっと面白いのがESCキーを押すと現れるポップアップメニューだ。

深い階層のメニューを，目的のコマンドを求めて降りていき，たどりつくと，おもむろに「始点を指定してください」。

まあ，それはいい。問題は，プリセレクションのときだ。普通の感覚では左ボタンでドラッグするのだが，“Multiword”はひと味もふた味も違う。左ボタンでドラッグ（とはいっても，指定範囲が反転してくれないので気分が出ないが）すると，ドラッグのボタンを放したとき，「範囲の先指定」という名のポップアップメニューがぽよんと現れる。なんとそこには，文字修飾とか

❷ウィンドウモードの画面

❹グラフィックエディタで描く

罫線とかのメニューがあるのだ。もし文字列のカット＆コピーをしたいなら，右ボタンでドラッグせねばならないのだ。すると，ちゃんと文字関係のポップアップメニューが登場する（図２）。

ユニークというかなんというか，なんといおう。不思議なふわふわ感である。

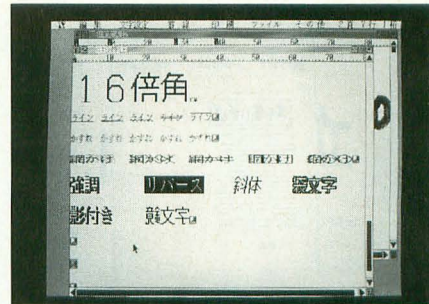
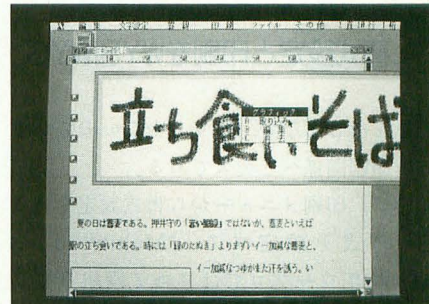
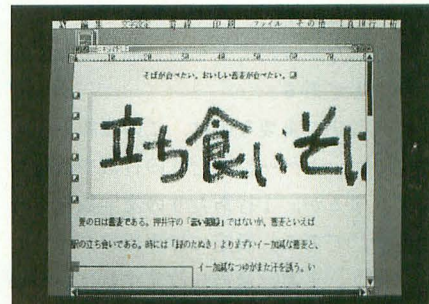
不思議なふわふわ感はまだある。“Multiword”はうれしいことに，ウィンドウのアイコン化が可能なのだ。じゃまなウィンドウはアイコン化してしまえばじゃまにならない。不思議なのは，アイコンを復元するときだ。普通の感覚では，そのアイコンをダブルクリックする。しかし，“Multiword”にダブルクリックという概念はない。じゃあ，どうするか，というと，右クリックなのであった。不思議である。

4.グラフィック作成モード

ウィンドウ画面において，メニューのその他から「グラフィック」を選択し，「新規作成」を選ぶと，グラフィックエディタが起動する。すると，作成する絵の大きさを聞いてくる。「おいおい，違うだろ」と思う。印刷するのが目的なら，作成するグラフィックの大きさを画面上のドットで決めるのではなく，印刷時の大きさ（単位はもちろん，cmかインチだ）で決めるべきである。まあ，いい。

そのあと，マジックパレットのサブセッ

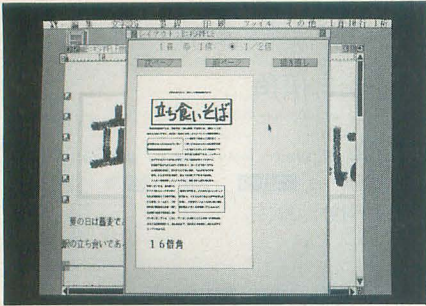
❸いろいろな文字修飾が使える

❺それを持っていって……

❻文書に取り込む

ト的なエディタがロードされる。サン・ミュージカル・サービス製のグラフィックエディタだ。ここで作った絵は，トリミングしたり大きさを変更したりして文書に取り込むことが可能である（写真４，５，６）。

ただし，このグラフィックモードは，あくまでもビットマップグラフィックスを文書に挿入する目的のものである。ビジネス文書や企画書にありがちな，矢印を書いたり，図形を書いたりして図示するようなものにはあまり向かない。

図２　範囲の先指定メニュー

左ドラッグ	右ドラッグ
A(1)文字飾り	A(1)カット(文字)
B(2)ノーマル	B(2)カット(行)
C(3)外枠罫線	C(3)コピー(文字)
D(4)罫線カット	D(4)コピー(行)
E(1)罫線コピー	E(5)貼付(カット)
F(2)罫線貼付(カット)	F(1)貼付(コピー)
G(3)罫線貼付(コピー)	G(2)検索
H(4)線種変更	H(3)置き換え
I(1)通過罫線	I(4)行ぞろえ
J(2)枠内操作	J(5)再変換
K(3)移動状態	
L(4)罫線モード	
M(1)罫線色変更	
N(2)罫線領域解除	

❼レイアウト表示

5.書式設定と印刷をWYSIWYG

そんなこんなで，残りも少ないのでいよいよ印刷関係へと突入である。

まず，印刷メニューから書式設定を選択し，書式を設定する。“Multiword”はX68000用ワープロではじめて段組みや袋とじが可能になっている（とはいえ，誰も覚えていないが，いちおうEWでもできないことはなかったことをつけ加えておく)。いいことである。

書式設定で重要なのは，この段組みをどうするか，1ページあたりの行数や改行幅をどうするか，1行あたりの文字数をどうするか，である。設定には専用のダイアログを使う。まあ，適当に設定すればいいわけだが，いちおう，WYSIWYGをうたっているだけのことはあるかどうかはチェックしておかねばならない。

清く正しいWYSIWYGは無理だとしても，最低限のことはやってほしいと思うからだ。それはなにか。ウィンドウ画面で，正しいバランスで表示されるかどうかである。つまり，設定した行間や字間が画面に反映されるかどうか，だ。

行間に関しては，書式設定で改行幅を広くすると，ちゃんと画面上でも行間があいた。これはよいことである（ただし，あまりあてにはならない)。しかし，字間は常に一定だ。残念である。

とにもかくにも，文書ができたら印刷する。いちおうレイアウト表示モードが用意されているので，レイアウトを見なければならない。レイアウト表示を実行する。待つ。……待つ。レーザープリンタならもう印刷が終わっているのではないか，というくらい待たされたあと，写真7のような画面が現れる。

レイアウトを確認したら，印刷だ。

印刷といえば，縦書き印刷もサポートしている。WYSIWYGと称していながらも，画面では横書きなのが，印刷すると，文字だけ縦になるという寸法なので，縦書きの場合はなおさら，レイアウト表示で確認する必要がある。

6.ファイル入出力

ファイル入出力のときには，シンプルなダイアログが現れる。ドライブ名の変更は手で行わなければならないが，あとは，マウスでもキーボードでも可能だ。

非常に面白いのが，文書のロードには2種類あるということ。現在開いているウィンドウに文書をロードする「文書ロード」コマンドと，新しくウィンドウを開いてそこにロードする「ロード＆オープン」コマンドだ。両者はまったく別のメニューにあるので，注意されたし。

ロードできる文書は，“Multiword”の文書のほか，テキスト，SWPファイル，JXWファイルの4つだ。JXWはいうまでもなく，一太郎 ver.3の文書ファイルである。

セーブできる文書形式はテキストと“Multiword”形式だけだ。泉大介氏に指摘されて気づいたのだが，「世の中のワープロの多くがJXWファイルの読み込みはサポートしているのに，JXWファイルの書き出しはサポートしていない」のは不思議である。少なくとも私は，「JXWファイルを読み込めなくてもいいから，JXWファイルを書き出したい」のである。JXW形式がワープロフォーマットのデファクトスタンダードとして通用しているのは事実である。つまり，文書ファイルをどこかへ持っていくとき，JXWファイルであれば，安全なのである。JXWファイルの書き出しをサポートすると，徳島の大企業が怒るのかもしれないが，んなのは無視してでも，やってほしかったと思う。

7.で，“Multiword”って何なんだ?

私はここまで“Multiword”を使って文章を書いてきた。絵も描いてみた。いままでぐちゃぐちゃとぼやいていたので，長所をまとめておこう。

○長所

段組みや縦書き印刷ができる

指定範囲のインデントが変えられる（重要！）

ショートカットキーのカスタマイズ

“Multiword”はなんということか，初心者向けに見える外見とは裏腹に，コントロールファンクションやファンクションキーなどなどの機能をカスタマイズできるのだ。

KEYTABLE.MWPというファイルを書き換えればかなり自由なカスタマイズができる。というより，カスタマイズしないと，「んがー！」な目に合う。デフォルトでは，CTRL＋Hにチャイルドプロセスが割り当てられているのだ。Human68kの頭文字がHだったばかりに，こういうことになる。CTRL＋Hにはやはりバックスペースというものだ。

だから，まず最初にコントロールファンクションを整備するのだ。もちろん，ESCファンクションとか，2段階のコントロールファンクションという気のきいたものはないから，考えて決める。

しかし，限界はある。恐ろしいことに，カーソルをその行の行末に飛ばす，という機能がない。行末ジャンプという機能は「その行が途中で終わっていても，強制的にいちばん右端まで飛ばしてしまう」ことなのだ。なんということだ。これでは一太郎ではないか。いや，一太郎でも，「その行に書かれた文の末尾」に飛ぶ機能は行末ジャンプとは別に持っているというのに。これではまったく意味がない。ちょっと前に戻って修正したあと，書きかけの行の続きを書きたい，というときは，カーソルキーでえっちらおっちらやるしかないのだ。

ユニークなASKのカスタマイズ

ありがちなことだが，FIXERは使えない。許しがたいことである。しかし，ASK専用だけあって，メニューから辞書登録できたり，インライン変換ができたり，というメリットはある。

さらにありがちなことに，ASKの操作性が拡張されている。文字が入力されていない状態で，XF2キーがローマ字のON/OFF，XF1キーがかな入力のON/OFFに割り当てられているのだ。さらに，コントロールファンクションにローマ字キーや全角キーを割り当てることもできる。

このXF1,XF2キーだけは便利だ。ASKが標準で持っていてもいい機能だ。

さらに，ウィンドウモードでは，写真のように候補の派手な一覧表が出るようになっている。

装飾が豊富
　レーザープリンタをサポートしている
　テキストモードがついている
　罫線モードが充実している
　罫線を使った表の右や左にも文書を入れることができる。
　罫線枠で編集ができる
　グラフィックエディタがついている
　任意倍角をサポートしている
　一太郎の文書ファイルを読める
　コントロールファンクションのカスタマイズができる

○短所は書かない

マウスを使ったマルチウィンドウにもかかわらず，コマンド体系がテキスト主体のDOSワープロを（わざと）真似しているところとか，範囲指定のしかたとか，キーボードで全操作可能とか，短所かどうかわからない点も多い。インデントが指定できるのはいいけれど，その行の左端と右端がルーラーに表示されないから，設定がどうなっているかわからない（これではルーラーの意味がないではないか）という点もある。

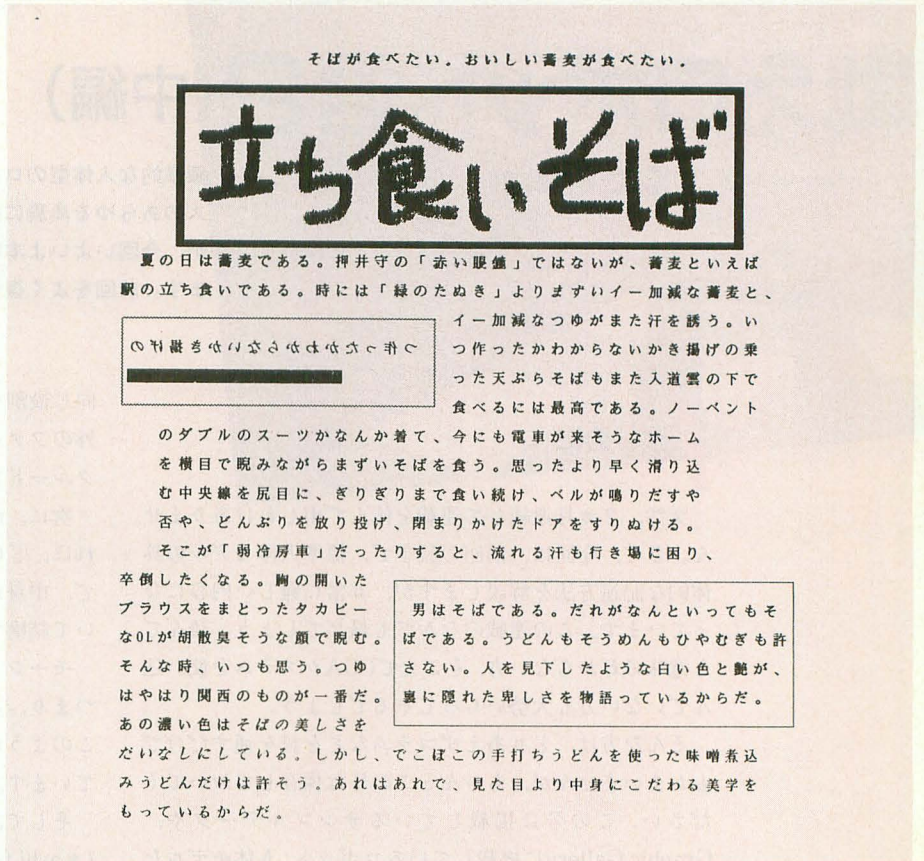
いまの世の中，純粋に，クリーンルームにこもってワープロを開発するなど無理な話である。ということは，設計者もいろんなワープロを見ているはずだし，いろんなワープロのおいしいところを参考にしているはずである。わからないのは，どれを参考にしたのか，である。この不思議な操作体系はどこから来たのか。たかだか検索をするのに，

“編集→文字列操作→検索”

と，3ステップも要する不思議な操作体系である。

少なくともMacintoshやWindows 3.0（英語版しかないけれど）のワープロをまったく参考にしていないのはたしかである。テキスト画面については，MS-DOSのワープロを参考にしたのはたしかである。

あ，もしかして，使ったことないからわからないけど，書院だったりして（笑）。

私は仕事柄いろんなマシンのいろんなワープロソフトを触る。それどころか，アバウトな性格をしているものだから，その時々によってPC-9801のVZエディタとVJE-βやMS-Worksで原稿を書いたり，DynaBookと一太郎dashで原稿を書いたり，ときにはMacintosh（EgBridgeとマックライトIIやSoloWriterという組み合わせだ）で書いたり，X68000でFIXER4.0とmicroEMACSで原稿を書いたり（この組み合わせがいちばんお気に入りである），ASKとHyperwordで書いたりする。そのくらいいろんなマシンとソフトを使う。

しかし，である。“Multiword”で書いた今回ほど「んがーっ」となったことはなかった。そりゃあ，一太郎dashを使っていて「んがー」というのは茶飯事だが，私はジャストシステムには何も期待していないけど，“Multiword”には少しは期待しているのである。だからだ。

もっとも，私のワープロの使い方は，一般人とは離れている。普通の人は私のように月に何十枚も原稿を書いたりしない。だから，少しくらい「んがーっ」でも，思ったような出来上がりの凝った文書が作れればそれでいいだろう。そういう人には，この多機能ぶりは気持ちいいに違いない。

私も「んがーっ」がおさまったら，「大人のためのX68000」の一環として，“Multiword”で複雑な文書を作る方法などもやってみたいと思っていたりする。

“Multiword”は文字どおり，マルチであることだけはたしかだからだ。

図3　印刷例

そばが食べたい。おいしい蕎麦が食べたい。

立ち食いそば

夏の日は蕎麦である。押井守の「赤い眼鏡」ではないが、蕎麦といえば駅の立ち食いである。時には「緑のたぬき」よりまずいイー加減な蕎麦と、イー加減なつゆがまた汗を誘う。いつ作ったかわからないかき揚げの乗った天ぷらそばもまた入道雲の下で食べるには最高である。ノーベントのダブルのスーツかなんか着て、今にも電車が来そうなホームを横目で睨みながらまずいそばを食う。思ったより早く滑り込む中央線を尻目に、ぎりぎりまで食い続け、ベルが鳴りだすや否や、どんぶりを放り投げ、閉まりかけたドアをすりぬける。

つ作ったかわからないかき揚げの

そこが「弱冷房車」だったりすると、流れる汗も行き場に困り、卒倒したくなる。胸の開いたブラウスをまとったタカビーなOLが胡散臭そうな顔で睨む。そんな時、いつも思う。つゆはやはり関西のものが一番だ。あの濃い色はそばの美しさをだいなしにしている。しかし、でこぼこの手打ちうどんを使った味噌煮込みうどんだけは許そう。あれはあれで、見た目より中身にこだわる美学をもっているからだ。

男はそばである。だれがなんといってもそばである。うどんもそうめんもひやむぎも許さない。人を見下したような白い色と艶が、裏に隠れた卑しさを物語っているからだ。

期待が大きすぎるのか

“Multiword”は人々の期待を一身に背負って登場した超多機能ワープロである。操作体系がちょっと不思議であるが，私のように，MacintoshだのWindows3.0だのといろいろ触りまくっているから，操作体系が不思議に思えるだけで慣れてしまえば気にならないのかもしれない（が，責任は持てない）。機能はたくさんあるし，遅いウィンドウ画面がいやなら，テキスト画面で文書を作り，ウィンドウ画面でレイアウトをすればいい。そんじょそこらのMS-DOSワープロよりは使えるだろう。数ページの見栄えがいい文書ならお手のものだ。

ちなみに，本レビューは製品バージョンより前の段階のものをシャープさんのご好意により使わせていただいたものである。よって，製品版と異なる点があるかもしれないのであらかじめご了承を。

総合評価（0　5　10）

項目	評価
処理速度	★★★★
Macintosh度	★★
MS-DOS度	★★★★★★★★★
WYSIWYG度	★★★
DTP度	★★★
グラフィックモー度	★★★★
多機能度	★★★★★★★
コストパフォーマンス	★★★★★★★★★

戦えロボット君2(中編)

プロジェクトチーム DōGA かまた ゆたか

標準的な人体型のロボットなどを自由に動かすため，フレームソースのあらゆる奥義に挑戦します。前回は簡単な構造体の場合でしたが，今回いよいよ本格的な人体モデルとなり，複雑さも極限に達します。前回をよく復習してからトライしてください。

はじめに

突然，2カ月連続して連載を休んで申しわけありません。さて，今回は，前回の続きで，標準人体モデルの具体的な記述方法を解説しますが，非常に難しい内容になっています。この連載のなかでも最高でしょう。読んでも意味のわからない方，そこまでCGAシステムを使い込んでいない方も大勢いらっしゃるでしょう。

そんな方は，とりあえずコラムなどを目を通すだけでもかまいませんが，ちゃんとこの号は保存しておいてください。この号に掲載しているサンプルデータや，Graphic Galleryに掲載しているロボット，人体モデルなどをまとめたサンプルデータ集を，近々“タケル”やパソコン通信で流そうと思っています。そのデータを受け取っていろいろ書き換えて遊ぶとき，必ずこの号が必要になるからです。

フレームソースファイルの構造

普通フレームソースは，ひとつのカットにつき，ひとつのファイルでしたが，標準人体モデルを使用する場合，7種類のファイルが必要になります。

図1をご覧ください。まず，通常のフレームソースと同じ役割をするのが，メインフレームソースで，それ以外のファイルはすべてこのメインフレームソースにインクルードされる形で利用されます。

次に，mandef.fscという定義ヘッダがありますが，これは，どのカット，どのモーションでも同じ内容ですので，中身がわからなくてもとにかく必要なんだと思っていて結構です。

モーションファイルは，ある動きの具体的なデータ，つまり，右股の関節は，このように傾いている状態から，このように変化するといった内容がずらずらと記述されています。

そして，体関数(＊.fsc)，腕関数(＊_ude.fsc)，足関数(＊ashi.fsc)，指関数(＊_yubi)という4つの関数ファイルは，各ロボット（標準人体型の多関節構造体）の構造を記述したファイルです。メインフレームソースから体関数を呼び，体関数が腕関数と足関数を呼び，さらに腕関数が指関数を呼ぶという複雑な構造になっています。

いきなりこれらすべてのファイルを理解しろというのは無茶です。そこで，次の3つのレベルに分けますので，各自ご自分のレベルと必要性に応じて習得していってください。

[レベル1] ロボットの形状データも，モーションデータも，ネットやタケルから取ってきて，それらを使って，いろんなカットを作る

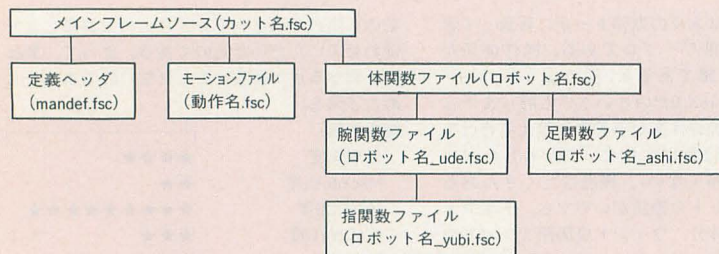
この場合，取り扱うのはメインフレームソースだけです。パターンさえ把握すれば，意味のわからないところが多少あっても，おまじないだと思って，気にせず作ることができます。

[レベル2] ロボットの形状データは，他人のを流用するが，動きだけは自分で作ってみる

モーションファイルだけ取り扱います。このモーションファイルの構造はとっても簡単なので，パターンはすぐ理解できます。しかし，具体的に各々のデータを求めるのはきわめて困難です。きっと近々よいツールが発表されることでしょう。

[レベル3] 新たにオリジナルロボットをデザインする

体，腕，足，指の各関数ファイルを取り扱います。これらのファイルはかなり複雑なのですが，どのようなロ

図1

ボットでも大部分が同じパターンになるので，それほど恐れる必要はありません。つまり，どのファイルにしてもパターンを理解してしまえば楽勝……そうは問屋がおろしません。最終的に本格的な作品を作るためには，これらを総合的に理解する必要があります。それでないと，エラーが出ても対応できませんし，思ったとおりの動きにならないとき，どこをどう直せばいいか見当もつきません。

いますぐにとはいいませんが，ちょっとずつ習得していってください。

メインフレームソース

リスト1(samp0.fsc)が通常のフレームソース，そしてリスト2(samp1.fsc)が人体モデルを使用したメインフレームソースです。例によって，いちばん左の行番号は便宜的につけたもので，実際に使用する場合にはつけないでください。

リスト1では，「ji(地面)」と「man」というひとつの物体を置いているだけ，リスト2では，「man」の代わりにtestという人体モデルを置いているだけです。2つのフレームソースを見比べてみると，ほとんど同じであることがわかります。人体モデルを使用する場合のメインフレームソースでも，視点や光源，人体モデル以外の物体は従来どおり記述します。

違っているところは，先頭になにやらわけのわからない数行(1～4行：宣言部)と「obj man」の代わりに入っている数行(19～22行：人体使用部)だけです。この2つの部分のパターンさえ理解すれば，一応メインフレームソースを作ることができます。

なお，宣言部にしろ，人体使用部にしろ，行の順番を変えるとエラーが発生する場合が多いので，順番もパターンに従ってください。

＊

1行目：#include "mandef.fsc"

「mandef.fsc」というファイルを取ってきて，その内容をこの行の位置にインクルード(書き加える)せよということです。「mandef.fsc」の内容はあと回しにして，人体モデルを使用するときは，無条件でこの1行を先頭に書くと覚えておけばいいのです。

2行目：#include "testfunc¥test.fsc"

再びインクルードですが，この行は各々のカットによって異なってきます。書式は，

#include "ロボット名.fsc"

となります。

「ロボット名.fsc」というファイルは図1の体関数のことです。リスト2の場合，ファイル名の前にパスを書いていますが，これは体関数，腕関数などは「testfunc」という別のディレクトリにまとめてあると仮定した場合であり，同じディレクトリにあればパスは必要ありません。

つまり，リスト2の例では，「test」という名のロボットをこのメインフレームソースのなかで使用すると宣言しているわけです。もし，「ZZ」という名のロボットを使用するなら当然，

#include "func¥zz.fsc"

となります。さらに，何種類かのロボットを使用する場合は，

#include "func¥test.fsc"
#include "func¥zz.fsc"
⋮

と複数行並べて書くことになります。

3行目：#include "motion¥rei.fsc"

モーションファイルをインクルードしています。一般的な書式は，

#include "動作名.fsc"

となります。モーションファイルの中身はあとで詳しく説明しますが，このファイルもひとつの関数の形式をしていますので，事前に宣言しておく必要があるわけです。この例でもパスが指定されていますが，これは，モーションファイルをすべて「motion」というディレクトリにまとめていることを想定しています。

また，このカットのなかで，複数のモーションを利用する場合も，

#include "motion¥jab.fsc"

リスト1 〔samp0.fsc〕

```
 1:#frame( fno, 1, 10 )
 2:@4.2@
 3:fram
 4:{
 5:    light pal( rgb ( 1.00 1.00 1.00 ) -3.00 -2.00 -4.00 )
 6:    { mov (  650 -300 300 )  eye deg( 60 ) }
 7:    { mov (    0    0 220 )  target }
 8:    { obj ji }
 9:    {
10:        mov  ( 0 0 0 )
11:        scal ( 5 5 5 )
12:        obj  man
13:    }
14:}
15:#endframe
```

リスト2 〔samp1.fsc〕

```
 1:#include "mandef.fsc"
 2:#include "testfunc¥test.fsc"
 3:#include "motion¥rei.fsc"
 4:#init     ¥ test_1[manmax] ¥
 5:
 6:#frame( fno, 1, 10 )
 7:@4.2@
 8:fram
 9:{
10:    light pal( rgb ( 1.00 1.00 1.00 ) -3.00 -2.00 -4.00 )
11:    { mov (  650 -300 300 )  eye deg( 60 ) }
12:    { mov (    0    0 220 )  target }
13:
14:    { obj ji }
15:
16:    {
17:        mov  ( 0 0 0 )
18:        scal ( 5 5 5 )
19:#define  pose test_1
20:#do      ¥ motion = div( 0, 100, 1, 10, fno ) ¥
21:#do      ¥ rei( pose[] , motion ) ¥
22:#do      ¥ test( pose[] ) ¥
23:    }
24:}
25:#endframe
```

#include "motion¥straight.fsc"

⋮

と複数行並べて書くことになります。

4行目：#init　¥ test_1 [manmax] ¥

一般的な書式は，

#init　¥ ロボット名＋@[manmax] ¥

となります。「#init」は配列の宣言を行うコマンドです。この配列は，モーションファイルからロボットのポーズのデータを受け取るために，あとで必要となります。＋@というのは，同一種類で，複数のロボットを扱うときに必要で，たとえば，イングラム型のロボットが3体出てくるカットの場合，

#init　¥ ingram [manmax] ¥

としてしまうと，3体とも同じデータになり，まったく同じ動きしか表現できません。ですから，1体ごとに配列を取り，

#init　¥ ingram_1 [manmax] ¥

#init　¥ ingram_2 [manmax] ¥

#init　¥ ingram_3 [manmax] ¥

というように，3行にわたって宣言するのです。この＋@の部分は，特に「_数字」にする必要はありません。各自が，そのカットでわかりやすいように記述してください。

人体モデルが複数の場合，2行目と4行目がややこしいので，ちょっとまとめておくと，

#include "ロボット名.fsc"

はロボットの種類だけ，

#init　¥ ロボット名＋@[manmax] ¥

はロボットの数だけ必要になるわけです。

人体使用部を解説する前に18行目にも少しご注目ください。

18行目：　scal (5 5 5)

文法的には，通常のスケール変換で，X，Y，Zの各軸方向に5倍しているだけです。前回の最後に触れましたが，標準人体モデルの身長はおよそ「100」であるという決まりになっています。ですから，実際に使用する場合は，そのカットのほかの物体とスケールが合うように，拡大縮小してやることになります。

それでは，人体部ですが，

19行目：#define pose　test_1

これもほとんど無条件に，

#define　pose　ロボット名＋@

と記述するだけです。＋@は当然，4行目の「#init ¥ ロボット名＋@[manmax] ¥」と同じものになります。「#define A B」というのは，“以後Aと書いてあるところを，Bに置換しろ”という意味です。

20行目：#do　¥ motion = div(0, 100, 1, 10, fno) ¥

この20行目は，“fnoが1のときmotionは0，fnoが10のときmotionは100になる”と書いてあるわけですが，いったいなんのことでしょう。

モーションファイルの中身を理解する必要はありませんが，すべてのモーションは，motion＝0の状態からスタートし，motion＝100の状態で終了するように作られています。

たとえば，普通に立っている状態から，椅子に腰掛けるというモーションでは，motion＝0が立っている状

リスト3 [mandef.fas]

```
 1: #func jab( pose[] , motion )
 2:      #do ¥ pose[move_x] = div( 0,   3, 0, 100, motion)¥
 3:      #do ¥ pose[move_y] = div( 0,   2, 0, 100, motion)¥
 4:      #do ¥ pose[move_z] =  -9 ¥
 5:
 6:      #do ¥ pose[koshi_x] = div(  0,   -8, 0,  100, motion) ¥
 7:      #do ¥ pose[koshi_y] = div(-10,   -1, 0,  100, motion) ¥
 8:      #do ¥ pose[koshi_z] =  -50 ¥
 9:      #do ¥ pose[onaka_x] =   -5 ¥
10:      #do ¥ pose[onaka_y] = div(   5,   7, 0,  100, motion) ¥
11:      #do ¥ pose[onaka_z] =    0 ¥
12:      #do ¥ pose[dotai_x] = div(  -5,  -3, 0,  100, motion) ¥
13:      #do ¥ pose[dotai_y] = div(  20,  25, 0,  100, motion) ¥
14:      #do ¥ pose[dotai_z] =    0 ¥
15:      #do ¥ pose[atama_x] =  -23 ¥
16:      #do ¥ pose[atama_y] =   -5 ¥
17:      #do ¥ pose[atama_z] =   55 ¥
18:
19:      #do ¥ pose[lkata_x] = div(  -5,  55, 0,  100, motion) ¥
20:      #do ¥ pose[lkata_y] = div(  -5, -20, 0,  100, motion) ¥
21:      #do ¥ pose[lkata_z] = div( -10,   0, 0,  100, motion) ¥
22:      #do ¥ pose[llude_x] = div(  10,  65, 0,  100, motion) ¥
23:      #do ¥ pose[llude_y] = div( -10, -25, 0,  100, motion) ¥
24:      #do ¥ pose[llude_z] = div(  40,  40, 0,  100, motion) ¥
25:      #do ¥ pose[l2ude_x] =    0 ¥
26:      #do ¥ pose[l2ude_y] = div(-140,   0, 0,  100, motion) ¥
27:      #do ¥ pose[l2ude_z] =    0 ¥
28:      #do ¥ pose[lhira_x] =    0 ¥
29:      #do ¥ pose[lhira_y] = div( -20,   0, 0,  100, motion) ¥
30:      #do ¥ pose[lhira_z] =    0 ¥
31:      #do ¥ pose[lyubi_1] =   70 ¥
32:      #do ¥ pose[lyubi_2] =   70 ¥
33:      #do ¥ pose[lyubi_3] =   70 ¥
34:      #do ¥ pose[lyubi_4] =   70 ¥
35:      #do ¥ pose[lyubi_5] =   70 ¥
36:      #do ¥ pose[lyubi_w] =   10 ¥
37:
38:      #do ¥ pose[rkata_x] =    5 ¥
39:      #do ¥ pose[rkata_y] =  -20 ¥
40:      #do ¥ pose[rkata_z] =   50 ¥
41:      #do ¥ pose[r1ude_x] =    0 ¥
42:      #do ¥ pose[r1ude_y] =  -40 ¥
43:      #do ¥ pose[r1ude_z] =    0 ¥
44:      #do ¥ pose[r2ude_x] =    0 ¥
45:      #do ¥ pose[r2ude_y] = -100 ¥
46:      #do ¥ pose[r2ude_z] =    0 ¥
47:      #do ¥ pose[rhira_x] =   20 ¥
48:      #do ¥ pose[rhira_y] =    0 ¥
49:      #do ¥ pose[rhira_z] =  -20 ¥
50:      #do ¥ pose[ryubi_1] =   70 ¥
51:      #do ¥ pose[ryubi_2] =   70 ¥
52:      #do ¥ pose[ryubi_3] =   70 ¥
53:      #do ¥ pose[ryubi_4] =   70 ¥
54:      #do ¥ pose[ryubi_5] =   70 ¥
55:      #do ¥ pose[ryubi_w] =   10 ¥
56:
57:      #do ¥ pose[lmomo_x] = div(  20,  22, 0,  100, motion) ¥
58:      #do ¥ pose[lmomo_y] = div( -43, -54, 0,  100, motion) ¥
59:      #do ¥ pose[lmomo_z] = div(  20,   5, 0,  100, motion) ¥
60:      #do ¥ pose[lsune_x] =    0 ¥
61:      #do ¥ pose[lsune_y] = div(  84,  93, 0,  100, motion) ¥
62:      #do ¥ pose[lsune_z] =    0 ¥
63:      #do ¥ pose[lashi_x] = div( -12, -30, 0,  100, motion) ¥
64:      #do ¥ pose[lashi_y] = div(  -5,  15, 0,  100, motion) ¥
65:      #do ¥ pose[lashi_z] = div(  20,  30, 0,  100, motion) ¥
66:      #do ¥ pose[ltuma_x] = div(   0, -10, 0,  100, motion) ¥
67:      #do ¥ pose[ltuma_y] = div( -35, -45, 0,  100, motion) ¥
68:      #do ¥ pose[ltuma_z] =    0 ¥
69:
70:      #do ¥ pose[rmomo_x] = div( -30, -25, 0,  100, motion) ¥
71:      #do ¥ pose[rmomo_y] = div( -33, -34, 0,  100, motion) ¥
72:      #do ¥ pose[rmomo_z] = div(  15,   5, 0,  100, motion) ¥
73:      #do ¥ pose[rsune_x] = div(   0,  -5, 0,  100, motion) ¥
74:      #do ¥ pose[rsune_y] = div(  76,  71, 0,  100, motion) ¥
75:      #do ¥ pose[rsune_z] = div(  10,  26, 0,  100, motion) ¥
76:      #do ¥ pose[rashi_x] =    0 ¥
77:      #do ¥ pose[rashi_y] = div(   0,   8, 0,  100, motion) ¥
78:      #do ¥ pose[rashi_z] = div(  20,  20, 0,  100, motion) ¥
79:      #do ¥ pose[rtuma_x] =    0 ¥
80:      #do ¥ pose[rtuma_y] =  -40 ¥
81:      #do ¥ pose[rtuma_z] =    0 ¥
82: #endfunc()
```

態，motion＝100が腰掛けている状態，そしてmotion＝50では，その中間の中腰の状態になります。つまり，motionという変数は，その動作のパーセンテージを意味するのです。

ですから，この20行目の意味は，“１フレーム目から動作が始まり，10フレーム目で完了する”と指定したことになるわけです。

21行目：#do　¥rei(pose［］, motion)¥

「rei.fsc」というモーションファイルのなかに記述してある「rei」という関数を実行しています。この関数に“礼をする”モーションのデータが入っていれば，「test」という名のロボットが礼をするカットが出来上がります。

#do　¥動作名(pose［］, motion)¥

という書式になります。

22行目：#do　¥test(pose［］)¥

前回解説したように，「obj」を置く代わりに「test」というロボットの関数を呼びます。一般的な書式は，

#do　¥ロボット名(pose［］)¥

となります。

定義ヘッダ（mandef.fsc）

この定義ヘッダは，前述のとおり，内容が変わることはありませんので，リスト３が「mandef.fsc」そのものです。

「#define」がずら〜っと並んでいるだけですが，標準人体モデルの記述をわかりやすくするちょっとしたポイントになっています。「#define」というのは，メインフレームでも解説したように，置換を意味します。たとえば，

#define　koshi_x　4

というのは，以後「koshi_x」と書いてあるところを全部「４」に置き換えるという意味になります。でも，「koshi_x」が腰のＸ軸回転の値であることは容易に予想できますが，「４」とはいったいなんの数値なんでしょうか。

ここでちょっと前回を思い出してほしいのですが，関数を実行するときに，複数のパラメータが必要で，人体モデルの場合，なんと76個（前回は間違えて64個と書いたような気もしますが）にも及びます。これらのパラメ

各読者連絡事項

□愚か者め

第３回CGAコンテストビデオは届きましたか？

発送作業が手間取り，若干遅れてしまいましたが，申し込んだけどまだ来ていないという方はご連絡ください。以下“愚か者”特集です。身に覚えがある人は，反省してから連絡してください。

・５月22日に佐賀県佐賀郡から申し込んだ名なしさん

・５月23日に群馬県多野郡から申し込んだ名なしさん

・５月20日に東京都文京区から申し込んだ名なしさん

・５月29日に神奈川県川崎市から申し込んだ名なしさん

・５月27日に大阪府大東市から申し込んだ名なしさん

・４月５日に国包から申し込んだ名なしさん

・５月28日に京都百万弁から申し込んだ藤川さん

以上，住所，氏名を明記されていない方です。国包の名なしさん，払込人住所氏名の欄に当方の住所を書いてはいけませんよ。

・広島県福山市の中田さん。5,000円もいただいたのはありがたいですが，いったい何をお求めですか？　CGAシステムですか，ビデオですか？　全額カンパとしていただいてよろしいでしょうか？

・新田町の亀井さん，新田町とはどこの県にあるのか教えてください。ついでに何をお求めなのかも教えてください。

・宮城県刈田郡の杉内さん，ビデオはVHSのみっていったでしょ。βはありません。どうしましょ。

・横浜市港北区の山中さん，発送したビデオが戻ってきています。正確な住所を教えてください。

・最後に極めつけ。香港の陳さん！　いくらなんでも香港まで2,000円でビデオは送れません。沖縄の最南端まで発送しておきますから，ご自分で取りに来てください！

□全日本X68000芸術祭について

すでにご存じのとおり，XVIのデビューを記念して，山下氏がプロデュースするコンテストが行われています。すでに，四国，北海道などの地区の予選が始まっておりますが，グラフィックス部門もありますので，CGAシステムを使われる皆さんも積極的にエントリーしてください。

この芸術祭に関して何件か問い合わせがありましたので，まとめて答えさせていただきます。

Ｑ１：CGAシステムで制作した作品でエントリーするのに，DōGAの了承が必要ですか？

Ａ１：まったく必要ありません。どんどんエントリーしましょう。

Ｑ２：DōGAが主催する「CGAコンテスト」にも応募したいのですが，２重エントリーは可能でしょうか？

Ａ２：CGAコンテストはほかのコンテストに応募した作品でも問題ありませんが，芸術祭の応募要項には，未発表の作品となっているので問題が生じます。しかし，山下氏と相談した結果，来年２月に行われる補選を除けば，応募締め切りがCGAコンテストより早いので，その時点では未発表であると解釈し，芸術祭とCGAコンテストへの２重エントリーを認めるという方針になりました。

Ｑ３：芸術祭への応募はディスクでということですが，CGA作品の場合枚数が多くなってしまいますが……。

Ａ３：この件も山下氏にお願いして，CGアニメーション作品の場合，ビデオテープでの応募も受け付けていただけることになりました。しかし，あまりに凄い映像を作りすぎると，本当にパソコンで制作したものかどうか区別ができませんので，ディスクに画像データや形状データとそれを表示できるプログラム（SLIDE.XやCAD.X）なども入れて，いっしょにエントリーしてください。

＊

当チームのスタッフにも，エントリーを計画している者がおりますので，大阪地区予選でご覧いただけるかもしれませんね。内容も教えてもらったのですが，まぁ全国大会ではご覧いただけないことは保証します!?　お楽しみに。

□CGA上映会in名古屋

いままで，東京，横浜，大阪，福岡などで上映会を行ってきましたが，金のシャチホコに賭けて名古屋でも上映会をしてほしいとの声に応えて，下記のとおり上映会を行います。３回のCGAコンテストのなかの名作や，なかなかお目にかけることができないデモなどを，解説付きでご覧いただきたいと思います。“安井百合江さん（17）X68000 PRO 愛知県”待ってろよ！（個人名を出すなって）

日時：1991年８月25日（日）

第１回 13：00〜　第２回 15：00〜

＊上映時間約１時間30分

場所：コムロード［地下鉄名城線上前津駅下車30秒，駅のすぐ上です］

ータをずらずらと記述するのはあまりに非現実的ですので，この76個のパラメータをひとつの配列（たとえば「test_1［］」）にまとめてしまいます。

1番目 2番目 3番目 4番目 ………… 75番目

test_1［］＝ move_x move_y move_z koshi_x …… rtuma_z

つまり，「4」とは，パラメータの配列の4番目を意味しているのです。

なぜ，こんなことをするかというと，

test_1［4］＝30

と表記するより，

test_1［koshi_x］＝30

と書くほうが，腰をX軸回りに30度ひねったということが，ずっとわかりやすいからです。そして，実際にFF（フレームファイルフィルタ）が実行されるときは，「mandef.fsc」によって，「koshi_x」が「4」に置換されるという仕組みです。

モーションファイル

リスト4がモーションファイルの一例で，左ジャブを打っているところです。

このファイルの形式も非常に単純で，「#do」文の羅列となっています。

#do ¥ 変数 ＝ 数値 or 数式 ¥

は，変数へ代入する書式で，「pose［］」という配列に，具体的なデータを代入しているのです。

1行目は関数の定義で，書式は，

#func 動作名(pose［］，motion)

となり，最終行の，

#endfunc( )

と対になっています。

さてモーションファイルは，メインフレームソースと違って，パターンだけを理解しても，具体的なモーションデータを作ることはできません。実際にモーションデータを作れるようになるためには，ちゃんと意味を理解する必要があります。

基本は，動作というものを，あるポーズからあるポーズへの変化と考えて，キーフレームになるポーズの各関節の各軸回りの角度を求め，「div」を使って補間してやります。たとえば，“首を横に振る”という動作の場合，右方向を見た状態と左方向を見た状態がキーフレームとなり，頭のZ軸の値だけが変化します（図2）。

#do ¥ pose［atama_x］＝ 0 ¥
#do ¥ pose［atama_y］＝ 0 ¥
#do ¥ pose［atama_z］＝ div(−30, 30, 0, 100, motion) ¥

このとき，メインフレームソースの解説にあったように，ひとつの動作はフレームナンバーの関数ではなく，「motion」の関数となります。そして，motion＝0のときがその動作の始まりで，motion＝100のときその動作が終わるようにします。

ですから，“首を横に振る”という動作は，“最初は正面を向いていて，それから首を右に，左に振り，最後にまた正面を向く”とすれば，

#do ¥ pose［atama_z］＝ div(0, −30, 30, 0, 0,25,75,100, motion) ¥

となります。

各関節の名前は，前回も掲載した図3をご覧ください。たとえば，右ヒジなら「r2ude」で，右ヒジのY軸回転なら「r2ude_y」となります。しかし，よく見るとモーショ

リスト4〔jab.fsc〕

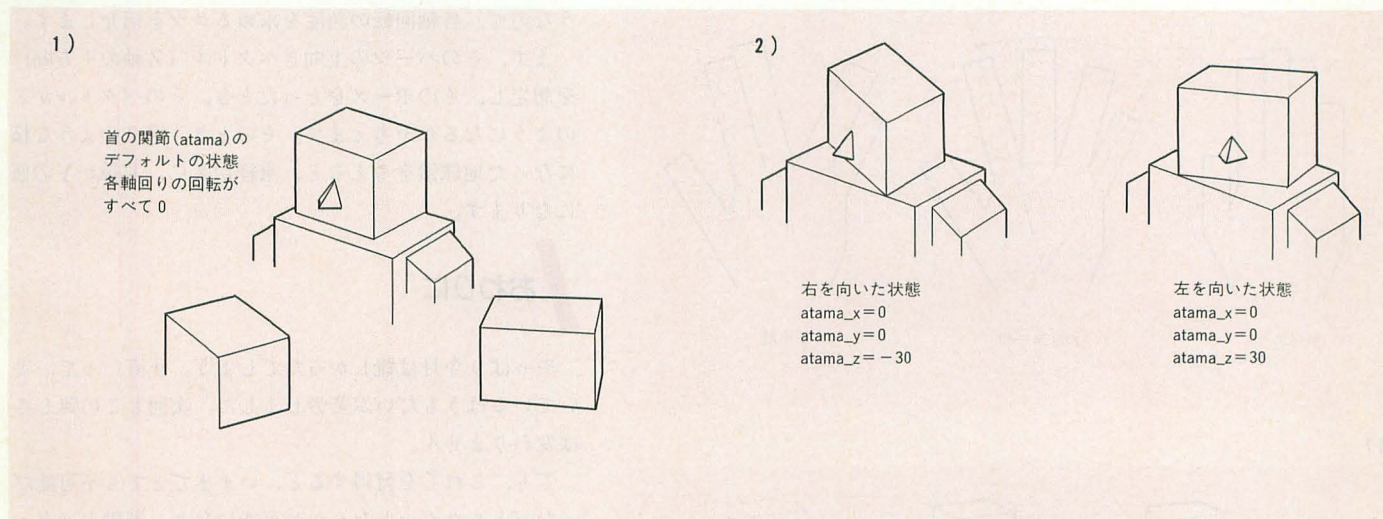
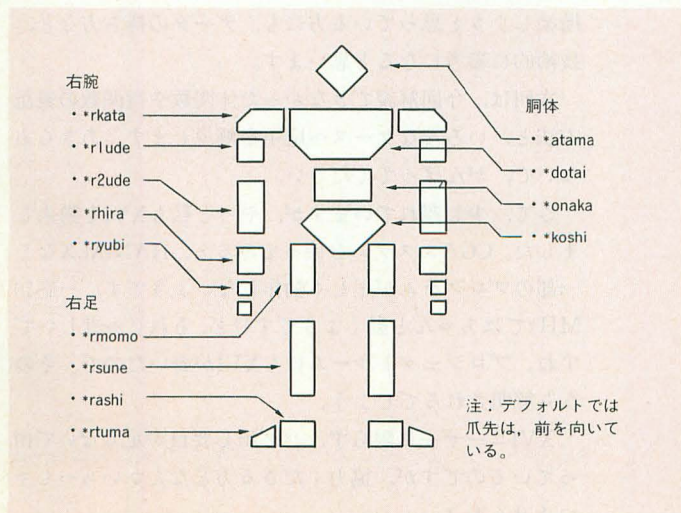
```
#define  move_x   1
#define  move_y   2
#define  move_z   3

#define  koshi_x  4
#define  koshi_y  5
#define  koshi_z  6

#define  onaka_x  7
#define  onaka_y  8
#define  onaka_z  9

#define  dotai_x  10
#define  dotai_y  11
#define  dotai_z  12

#define  atama_x  13
#define  atama_y  14
#define  atama_z  15

#define  lkata_x  16
#define  lkata_y  17
#define  lkata_z  18

#define  llude_x  19
#define  llude_y  20
#define  llude_z  21

#define  l2ude_x  22
#define  l2ude_y  23
#define  l2ude_z  24

#define  lhira_x  25
#define  lhira_y  26
#define  lhira_z  27

#define  lyubi_1  28
#define  lyubi_2  29
#define  lyubi_3  30
#define  lyubi_4  31
#define  lyubi_5  32
#define  lyubi_w  33

#define  rkata_x  34
#define  rkata_y  35
#define  rkata_z  36

#define  rlude_x  37
#define  rlude_y  38
#define  rlude_z  39

#define  r2ude_x  40
#define  r2ude_y  41
#define  r2ude_z  42

#define  rhira_x  43
#define  rhira_y  44
#define  rhira_z  45

#define  ryubi_1  46
#define  ryubi_2  47
#define  ryubi_3  48
#define  ryubi_4  49
#define  ryubi_5  50
#define  ryubi_w  51

#define  lmomo_x  52
#define  lmomo_y  53
#define  lmomo_z  54

#define  lsune_x  55
#define  lsune_y  56
#define  lsune_z  57

#define  lashi_x  58
#define  lashi_y  59
#define  lashi_z  60

#define  ltuma_x  61
#define  ltuma_y  62
#define  ltuma_z  63

#define  rmomo_x  64
#define  rmomo_y  65
#define  rmomo_z  66

#define  rsune_x  67
#define  rsune_y  68
#define  rsune_z  69

#define  rashi_x  70
#define  rashi_y  71
#define  rashi_z  72

#define  rtuma_x  73
#define  rtuma_y  74
#define  rtuma_z  75

#define  manmax   76
```


緊急のお知らせ
8月22日（木）～8月28日（水）AM10：00～PM7：00に，広島県広島市サンモール5階 YYYプラザにてCG研究会，サンモール主催で「CGアート展」が行われることになりました。皆さまのご来場をお待ちしています。

ンファイル（リスト4）には，図3にはない変数がいくつかあります。

まず最初の「move_x」，「move_y」，「move_z」の3つは，その動作による腰の微少な位置のずれを補正するために与えます。

たとえば，“礼をする”という動作の場合，単純に上半身だけ曲げると，重心が足より前になり，前に倒れてしまいますので，重心がちゃんと足の位置にくるように腰を後ろにさげたほうが自然です（図4）。

この移動量をメインフレームソース内で記述すると，いろんな動きを利用するたびに，その人体モデルの位置を微妙に変化させる必要が出てきます。そこで，その動作を行えば，必ず発生する位置の変化はあらかじめモーションファイル内に記述してやるのです。このとき，同じ“礼をする”という動作でも，大きなロボットと，小さなロボットでは位置の変化量が変わってしまうのですが，標準人体モデルは身長が「100」という決まりがありますので，身長に対して何パーセントぐらい変化したかという値を記述することになります。

そして次は，まん中へんにある，「lyubi_1」～「lyubi_5」，「lyubi_w」と，「ryubi_1」～「ryubi_5」，「ryubi_w」ですが，これらは，指の状態を記述する変数です。なお，1文字目の「l」と「r」はもちろん左手と右手です。

手は非常に多くの関節からなっており，これらすべての関節において，体のほかの関節同様にX，Y，Z回転の値を用意していてはあまりに複雑になりすぎます。そこで，X，Y，Z軸による表記を捨て，各指の状態を表す5つの変数と，指と指の間の広げ具合を表すひとつの変数で表現します。

	yubi_1	yubi_2	yubi_3	yubi_4
yubi_5	yubi_w			
	親指	人差指	中指	薬指
小指	広げ具合			

図5のように，変数の値がその指のすべての関節の角度となります。ですから，第一関節だけ曲げるというようなことは表現できません（そんな器用なことは私もできません）。「yubi_1」から「yubi_5」まで「0」ならパーに，「70」ぐらいならグーに，「yubi_2」と「yubi_3」

図2

1）

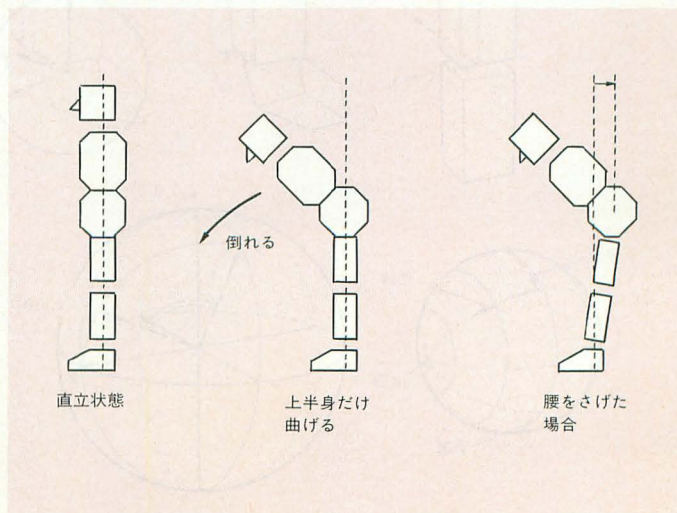
首の関節（atama）の
デフォルトの状態
各軸回りの回転が
すべて0

2）

右を向いた状態
atama_x＝0
atama_y＝0
atama_z＝－30

左を向いた状態
atama_x＝0
atama_y＝0
atama_z＝30

図3

図4

だけ「0」でそれ以外が「70」ならチョキになります。指が骨折したところを表現するのでもなければ，通常－5～90の範囲になります。

「yubi_w」については，図6をご覧ください。同じパーでも，敬礼のときは，「yubi_w＝0」であり，張り手のときは「yubi_w＝50」ぐらいになります。この場合も，人差し指と中指，薬指と小指をくっつけたまま広げるというエンタープライズ号のクルーのような表現はできません（私もできないんだって）。

図5

図6

図7

以上でモーションファイルの書式はご理解いただけたと思います。しかし，よほど簡単なポーズでない限り，各関節のX，Y，Z軸回転の値を求めることは，非常に大変な作業となるでしょう。3次元の軸回転では，回転の順番によって結果が異なります。この標準人体モデルでは，この回転の順番が，X，Y，Zと固定化されていますので，さらに難しくなっています。いままでに多関節構造体を扱ったことがある方なら，この制約が非常に問題のあるものであることがおわかりいただけるでしょう。

しかし，現実的に考えると，たとえこの制約がなくても，各軸回転の角度をエディタで1つひとつ書いていくことはあまり行われないでしょう。結局は，視覚的に，インタラクティブに，ポーズをデザインできるようなツールが必要になります。

現在そのようなツールを開発中ですが，皆さんのお手元に届けられるようになるまでには，まだ少しかかりそうなので，各軸回転の角度を求めるコツを紹介します。

まず，そのパーツの上向きベクトル（Z軸の＋方向）を想定し，そのポーズをとったとき，そのベクトルがどのようになるかを考えます。そのとき，図7のような横になった地球儀を考えると，東経がXに，北緯がYの値になります。

おわりに

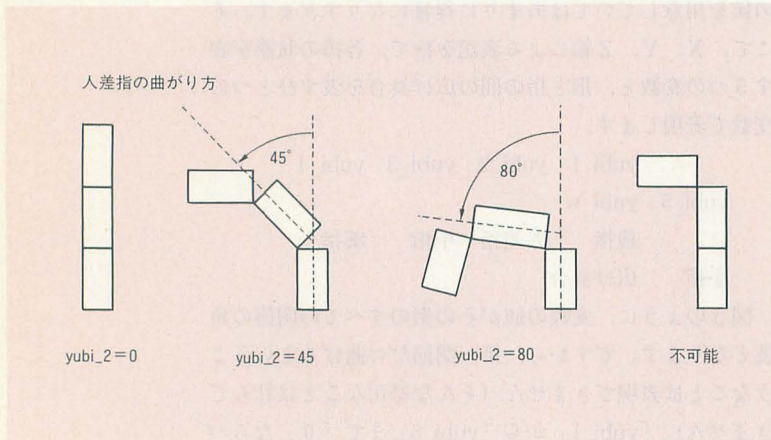
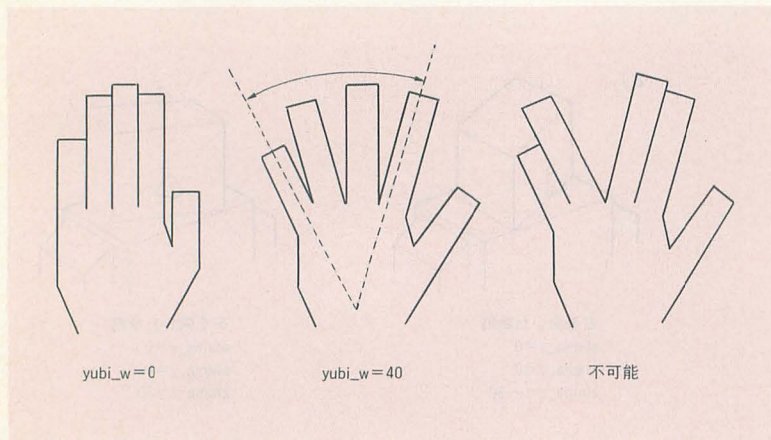
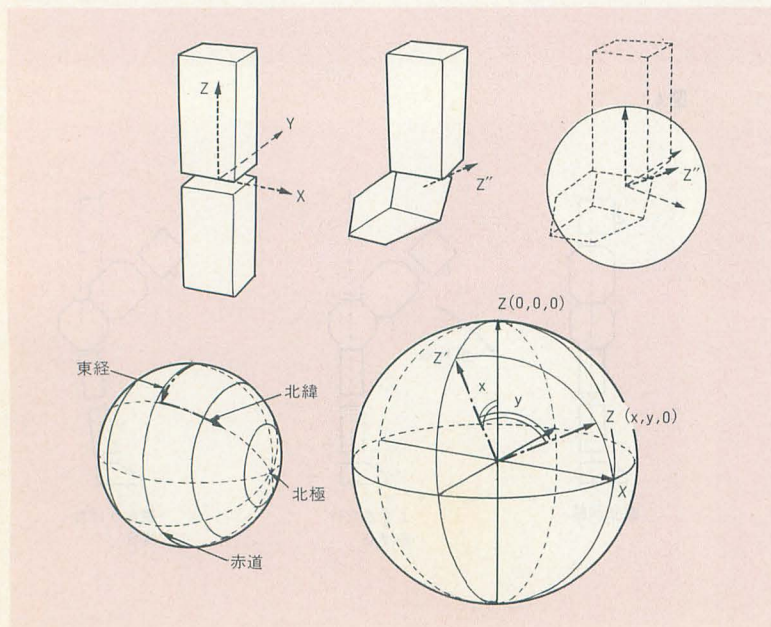
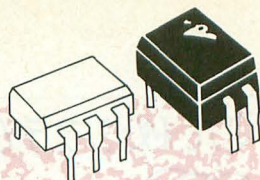
やっぱり今月は難しかったでしょう。正直いって，書いているほうもだいぶ苦労しました。次回もこの難しさは変わりません。

でも，これらを習得すると，いままでとても不可能だったバトルロボットなんかが可能になり，表現力はぐっとレベルアップします。またオリジナルのCGシステムを構築しようと思っている方にも，データの持ち方など，技術的に参考になると思います。

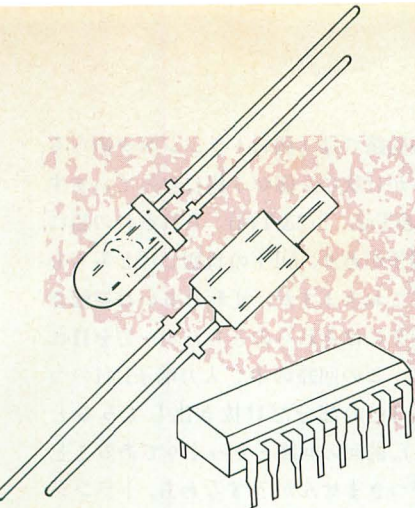
次回は，今回解説できなかった体関数や指関数の表記方法と，いろんなケースへ応用を解説します。あきらめないで，がんばってください。

さて，少し遅れていますが，やっと私もXVIを購入しました。CGAシステムを使ってみると，HANIM.Xなど一部のプログラムが正しく動作しないようです。一応10MHzではちゃんと動くようですけど，それじゃ悲しいですね。プロジェクトルームにもXVIが着いたので，そのうち解明されるでしょう。

XVIユーザーに限らず，バグ出し要員が足らないで困っているのですが，協力くださる方どなたかいらっしゃいませんか？

ハードウェア工作入門《15》

ハイテクタンク製作（理論編）

Misawa Kazuhiko
三沢 和彦

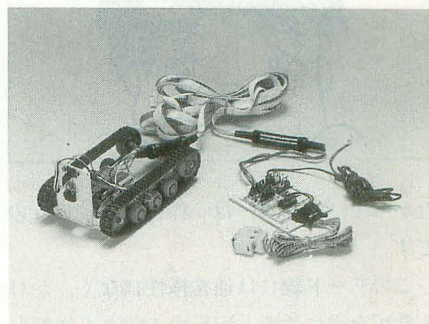
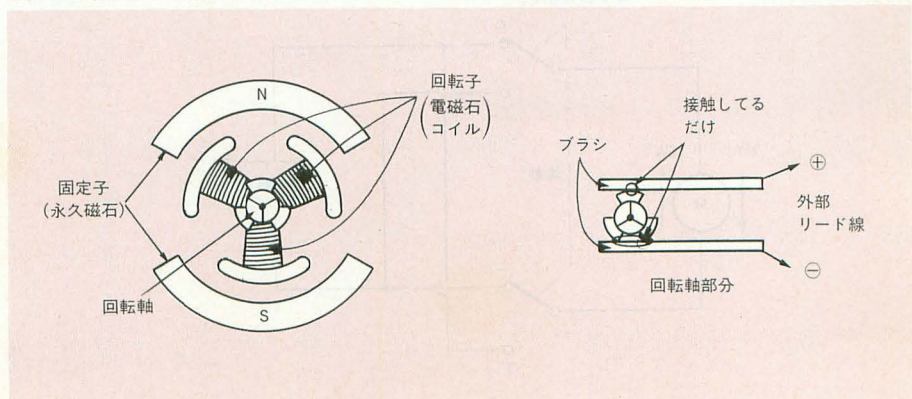
今月から直流モーターを使ったキャタピラタンクの製作を開始します。まずは理論編ということでモーター制御のインタフェイスをじっくり解説していきます。キャタピラの前後進制御に必要な正逆転制御ロジックをぜひとも理解しておきましょう。

模型用モーターの制御

前回のキャタピラタンクを編集部でデモンストレーションしたところ,「いも虫が這っているみたい！」と酷評を受けました。じっと我慢して見ているとやっと動いているのがわかるという，なんともお粗末なタンクでした。問題はステッピングモーターの回転速度が一般の模型用モーターの回転速度より遅いため，それをさらにギヤで減速してしまうとキャタピラの動くスピードが大変遅くなってしまうという点にありました。それに加え，モーター1個で左右両輪を駆動しているために曲がれないのです。もともとステッピングモーターというのは，位置決めを正確にすることが得意で高速にビュンビュン回す目的にはあまり使わないものです。

そこで，これらの欠点を克服すべく最高性能のキャタピラタンクを設計しました。名づけて，超高性能ハイテクタンク「パトリオット」。「パトリオット」といえば，湾岸戦争で多国籍軍が使用したミサイル迎撃用ミサイルの名前です。このネーミングの理由は今のところ秘密ですが前回少し予告したように，このタンクに搭載するマル秘システムに関係しているというヒントを出しておきましょう。なお，こちらの「パトリオット」は実際の戦争には使えません。私自身はアメリカ合衆国の湾岸戦争開戦には絶対反対でしたから。

さて，「パトリオット」の基本性能は高速の前後進および左右旋回です。その動力部分は模型用の直流モーターを使っています。今月と来月とでまず直流モーターをパソコンでコントロールするインタフェイスを設計，製作することにします。そして，足回りなど機械部分は再来月の応用編でじっくり設計，製作，実習することにして，さらにそれを操縦するプログラムを組みます。その次の月にはいよいよマル秘システムを公表し，その設計，製作に着手します。最終的にはこの「パトリオット」の性能をフルに生かすコントロールプログラムを完成させる予定です。完成までに理論編から実習編，応用編，発展編，完成編と全5回を予定しています。では，さっそくモーターのインタフェイス部分から始めましょう。

模型用モーターのインタフェイス

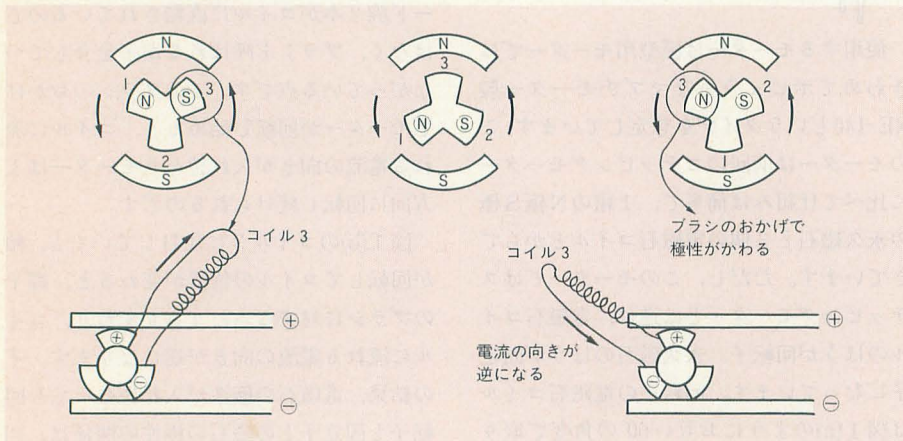
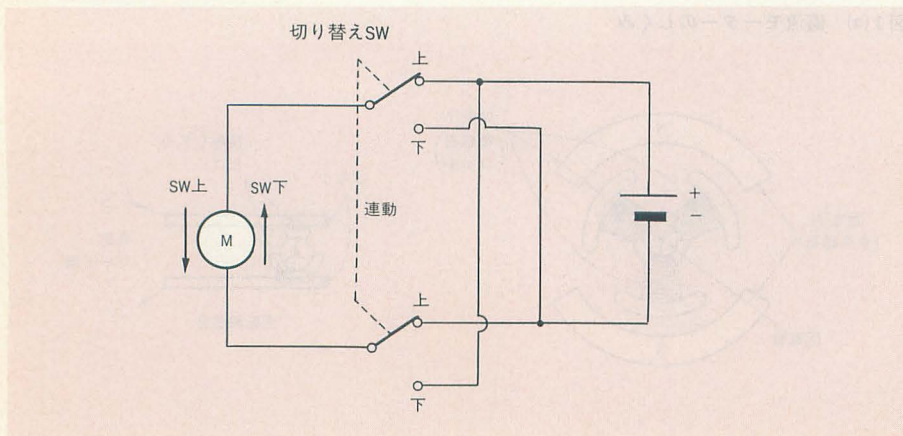
使用するモーターは模型用モーターではきわめてポピュラーなマブチモーター製RE-140というタイプを想定しています。このモーターは前回のステッピングモーターに比べて仕組みは簡単で，1組のN極S極の永久磁石と3個の電磁石コイルとからできています。ただし，このモーターではステッピングモーターとは逆に，電磁石コイルのほうが回転子，永久磁石のほうが固定子になっています。回転子の電磁石コイルは図1(a)のようにお互い60°の角度で取り付けられ，それぞれのコイルの端は回転軸上に図1(a)のように取り付けられた端子とつながっています。

ハイテクキャタピラタンクの完成写真

この種のモーターの特徴は，外からのリード線2本がコイルに直結されているのではなく，ブラシと呼ばれる端子を介してつながっている点です。このブラシのおかげでモーターが回転し始めると，コイルに流れる電流の向きが入れ替わりモーターは1方向に回転し続けられるのです。

図1(b)のコイル3に注目していくと，軸が回転してコイルの位置が変わると，端子のブラシに対する当たり方も変わり，コイルに流れる電流の向きが逆になります。その結果，電磁石の極性が入れ替わっても回転子と固定子との磁石の極性の関係は，回転中，常に変わらないことになります。し

図1(a) 直流モーターのしくみ

たがって，モーターは一様に回り続けるのです。

このリード線には通常極性はなく，かける電圧を逆にすると回転する向きも逆転します。そこで，リモートコントロールで電圧の向きを逆転するような回路を作ってやれば，コンピュータからタンクの前後進を操作することが可能となります。

ところで，極性を逆転するには，図2のような切り替えスイッチがいちばん簡単でしょう。ただし，モーターに少し大きめの電流を流さなければならないことを考慮しておく必要があります。すると，まず考えられるのはリレーという電気信号で制御できる機械的スイッチを使うことですが，これはあまりスマートな方法ではありません。というのも，リレーは電磁石を使った機械的スイッチでコイルに電流を流して動作させるという点がモーターと同じだからです。リレーのためのインタフェイスを作ってそこにモーターをつなぐぐらいなら，直接モーターを制御するインタフェイスを作ったほうが早いのです。

さて，コイルに流れる電流をON/OFFさせるのは，前回のステッピングモーターのインタフェイスと同様にトランジスタスイッチを使います。問題の正逆転回路もトランジスタスイッチをうまく組み合わせることで実現できます。図3にトランジスタスイッチ部分の基本的な回路図を示します。

この回路では4個のトランジスタのうち，たすきがけ状に2個が1組になっています(Tr1&Tr2，Tr3&Tr4)。それぞれの組に入力端子があり，OFFの状態ではどちらもLにしておきますが，正転あるいは逆転させるときにはそれらのどちらか一方をHにします。この回路の中で入力端子IN1のつながるTr1の部分だけ抜き出してみると(図4)，前回の回路とそっくりであることに気がつきませんか？ すなわち，トランジスタスイッチとしての基本原理はまったく同じで，異なるのは正逆転の切り替えができるように加えたTr2とダイオードです。そして，Tr3とTr4との組み合わせはTr1とTr2の組み合わせと同じで，回路図も左右対称になっていることがわかります。

回路図の中にあるTr2の記号をよく見るとこれまでのものと矢印の向きが逆になっています。これらのトランジスタの違いをまず説明しておきましょう。トランジスタには極性の違いから，図5のようにNPN型とPNP型の2種類があります。トランジスタ内のミクロな動作原理から説明していると混乱してしまうので，これらの違いを使い方の面からひと言でいうと，NPN型は入力がHでON，PNP型はLでONになる，ということです。

すなわちNPN型では，図5のような接続でベース端子を入力としたときに，コレクタ→エミッタ間に電流が流れるのは，ベース→エミッタ間に電流を流したときです。エミッタがGNDに接地しているので，ベースに正の電圧(H)をかけたときがON状態になります。それに対しPNP型ではエミッタ→コレクタ間およびエミッタ→ベース間に流れる電流の向きがNPN型と逆になっています。ですから，エミッタ→コレクタ間に電流が流れるのは，エミッタ→ベース間に電流を流したときです。こちらの場合ではエミッタが電源電圧に接しているので，ベースに電源電圧より低い電圧（L）をかけたときがON状態になります。

それでは図3の回路の動作を順を追って考えてみましょう。混乱したら図4の抜粋部分だけに注意して追ってみてください。入力端子IN1をHにすると，R1を通してTr1のベース→エミッタ間に電流が流れ，Tr1がONになります。するとTr1のコレクタ→エミッタ間が導通状態になりますから，

図1(b) モーターの回転

図2 切り替えスイッチ

R2を通してTr2のエミッタ→ベース間に電流が流れます。ここで，D1の役割はR2を通して流れてきた電流がモーターのほうへ回り込むのを防ぎ，Tr1のほうへ流してやることです。すると今度はTr2もONになるので，エミッタ→コレクタ間が導通状態になります。

さて，Tr2のコレクタ端子からはモーターとダイオードD2とにつながっていますが，D2の先にあるTr3はOFFなので，そちらには電流は流れません。それに対しモーターの先にあるTr1は，最初からONになっていますからモーターにはプラス端子からマイナス端子に電流が流れるのです。

今度は逆に入力端子IN2のほうをHにしたときの動作を各自追ってみてください。Tr1とTr2がそのままTr3とTr4とに入れ替わり，全体が左右対称になるだけですから簡単だと思います。全体に左右対称に電流が流れるわけですから，モーターにも逆向きに電流が流れ，その結果，逆回転することになります。

現在では，図3に示した正逆転対応トランジスタスイッチ全体をひとつのパッケージにまとめてしまったICも市販されています。一例として，沖電子から出ているTA7257Pを挙げておきましょう。しかし，これらのICはまだ特殊らしく，T-ZONEでも簡単には手に入らないということなので，今回は使用をあきらめました。余力のある人は自分で研究してみてください。

正逆転制御ロジックの設計

さて，実際のタンクでは左右旋回もできるようにします。そのためには，左右のキャタピラを独立に駆動できるようにモーターを2個取り付け，上で述べたインタフェイスも2組作ります。したがって，入力は全部で4端子あることになります。ところがX68000のジョイスティックポートは出力が3ビットなので，そのままでは直結できません。そこで，実際の動作を考えてみるとキャタピラの動作は前後進および左右旋回の組み合わせで，

(前進，後進)×(停止，直進，左，右)

の計8通りですむことがわかります。これはちょうど3ビット出力で表せますので，

IOC4：前後進

IOC6：左キャタピラ駆動

IOC7：右キャタピラ駆動

というように各ビットを割り当てることにします。すると，制御コードの組み合わせは，

IOC4：	IOC6：	IOC7：	
0	0	0	：停止
0	0	1	：左前進
0	1	0	：右前進
0	1	1	：直前進

図3　正逆転対応トランジスタスイッチ

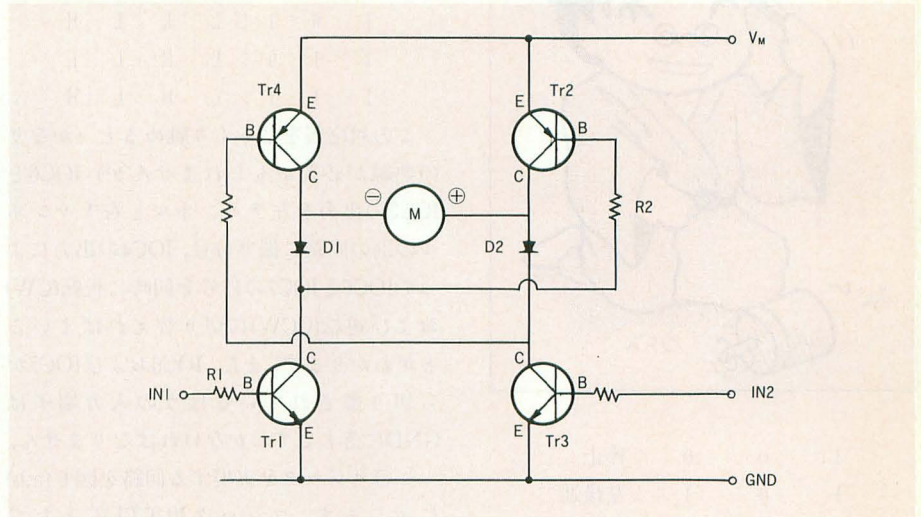



図4　トランジスタスイッチの基本部分

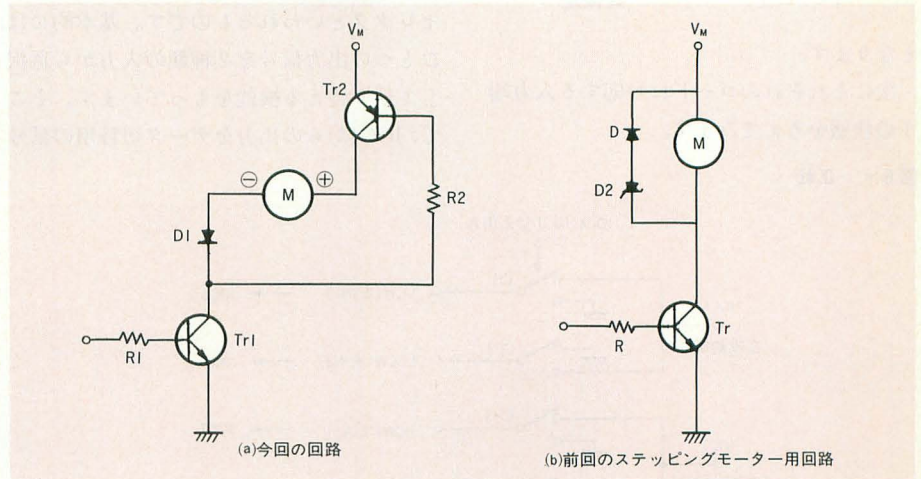
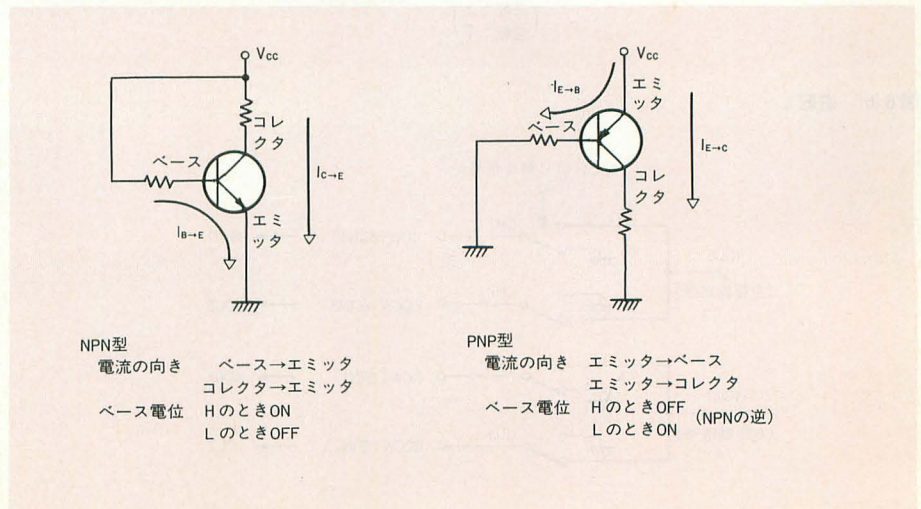



図5　NPN型トランジスタとPNP型トランジスタ

1	0	0	：停止
1	0	1	：左後進
1	1	0	：右後進
1	1	1	：直後進

となります。

次にそれぞれのコードに対応する入力端子の状態を考えてみます。

IOC4	IOC6	IOC7		RCW	RCCW	LCW	LCCW
0	0	0	：	L	L	L	L
0	0	1	：	L	L	H	L
0	1	0	：	H	L	L	L
0	1	1	：	H	L	H	L
1	0	0	：	L	L	L	L
1	0	1	：	L	L	L	H
1	1	0	：	L	H	L	L
1	1	1	：	L	H	L	H

この対応表をじっくり眺めると（かなりの熟練が必要かもしれませんが）IOC6とIOC7の出力を左チャンネルと右チャンネルの別の回路に振り分け，IOC4の出力によってIOC6とIOC7の信号を同時に正転(CW)および逆転(CCW)に切り替えればよいことがわかります。また，IOC6およびIOC7から切り離されているほうの入力端子はGNDに落としておかなければなりません。

このロジックを実現する回路を図6(a,b)に示します。ロジック用TTLICとしてLS157を使いました。これは，2→1データセレクタといわれるものです。基本的には，ひとつの出力信号を2種類の入力から選択して切り替える機能をもっています。そこで，IOC4からの出力をデータ切替用の信号として使うことにします。そして，正転用の入力端子と逆転用の端子とにデータセレクタの出力をつないでおいて，正転を選択したときには正転用の入力端子に駆動信号が加わり，反対の逆転用の端子はGNDに落ちるように配線するのです。言葉ではわかりにくいと思いますので順番に動作を追っていくことにしましょう。

まず，IOC4のセレクトが正転（上側）を選択したとします。このとき，データセレクタは図6(a)のように設定されています。逆転用の入力端子をセレクタのほうにたどっていくと（セレクタ2と4）GNDに設置されているほうに接続されていることがわかります。正転用の入力端子のほうはというと，左チャンネルがIOC6，右チャンネルがIOC7につながっているので，コンピュータから直進および左右旋回がダイレクトに制御できることになります。次にIOC4を逆転（下側）にすると，データセレクタの接続が図6(b)のように切り替わります。

今度は正転用の入力端子をセレクタのほうからたどっていくと（セレクタ1と3）GNDに設置されているほうに接続されることになります。その代わり逆転用の入力端子のほうについては，左チャンネルがIOC6，右チャンネルがIOC7につなぎ換えられ，こちらのほうがコンピュータから直進および左右旋回を制御できるようになるのです。

データセレクタの出力（図6）とトランジスタスイッチの入力（図3）は直結できるように設計してありますから，制御回路としてこれ以上のものは必要ありません。

図6(a)　正転

図6(b)　逆転

まとめ

以上でインタフェイス部分の設計は完成です。そんなに新しい事柄がなかったので理解できなかった人はほとんどいないと思います。もしも，トランジスタスイッチのところでつまずいたら，前回のステッピングモーターの解説とあわせて読んでみてください。特にロジック回路（データセレクタ）のところで混乱することもあるでしょう。そんなときはロジック回路の部分をプログラムと一緒に理解していけばよくわかると思いますので，次回以降も繰り返し説明していくつもりです。

吾輩はX68000である
［第5回］

最後の砦，配列と構造体

Izumi Daisuke　泉　大介

3回に及んだC言語の概略もそろそろ終焉を迎える
これで諸兄はC言語も扱えるようになったことだし
吾輩とのパソコンライフも一層楽しくなることだろう

C言語の概要も今回で一応の完結を迎える。先月までで諸兄は，C言語のプログラムが関数の集合体であること，関数の記述方法，C言語の制御構造をマスターしたことになる。すでに簡単なプログラムなら自力で作成することができるようになっているはずである。今回は，これまでに触れていなかった配列と構造体について，それがどのような形で実現されているのかをご覧に入れようと思う。

データの塊，配列

御仁は戯れにカレンダープログラムを作ることが少なくないが，このとき各月の日数をメモリに収めておき，表示しようとする年が閏年かどうかによって2月の日数をセットし直すという方法をよく使う。具体的には，図1のようにメモリの適当な番地から日数を順に収めておき，閏年ならB0001$_{H}$番地に29をセットし直すわけである。閏年の場合には，

```
move.b #29, $b0001
```

とすればいいことになる。

デバッガを使うなら，図1のようなデータの塊をメモリに作ることは簡単である。mesコマンドを使って1バイトずつデータをセットしていけばいい。各月の日数はたかだか31日どまりなので，1バイト(255まで表現できる)単位にセットしていけば十分である。では，C言語ではどうするか。これまでの知識を動員するなら，

```
char ＊days；
days=(char＊)0xB0000；
＊days=30；
＊(days+1)=29；
　　　：
```

としていけばいいことは簡単に想像できるだろう。「char型(1バイト)のデータが入っているアドレス」を保持するための変数daysを宣言しておき，アドレスとしてB0000$_{H}$をセット。そこから順に日数をメモリに書き込んでいくわけである。

しかし，これではいかにも面倒臭い。もっと簡単にやっつける方法はないものかとC言語を作った人は考えた(に違いない)。配列というデータ型がサポートされたのである。配列は，

```
char days1［12］；
```

のように宣言する。これはchar型のデータを12個入れるためのメモリを確保し，その先頭アドレスにdays1という名前をつけなさいという指示である。char型のデータ12個分のスペースがメモリのどこに確保されるのかは問題ではない。大切なのは，要求したスペースが確保され，それにdays1という名前がつけられるという点である。

変数は宣言するときにデータをセットしておくことができる。たとえば，

```
int test=10；
```

ならば，testという名前のint型データを入れるためのメモリが確保されると同時に，そのメモリに10がセットされる。配列も同様に，

図1 各月の日数のメモリ表現

B0000	31	28	31	30
B0004	31	30	31	31
B0008	30	31	30	31

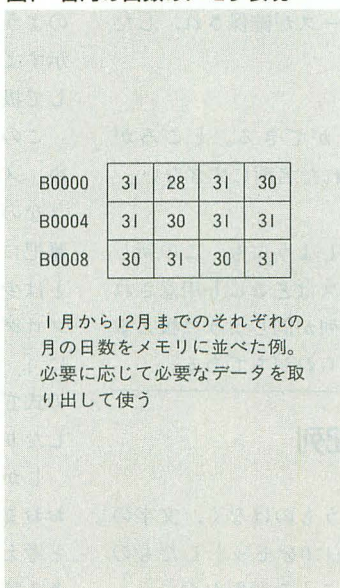
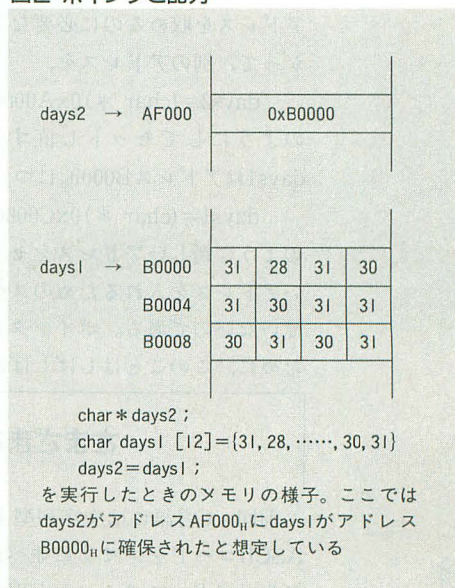
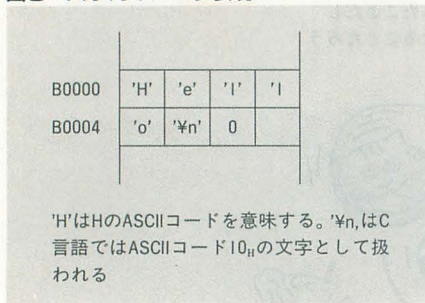
1月から12月までのそれぞれの月の日数をメモリに並べた例。必要に応じて必要なデータを取り出して使う

図2 ポインタと配列

```
char＊days2；
char days1［12］={31, 28, ……, 30, 31}
days2=days1；
```

を実行したときのメモリの様子。ここではdays2がアドレスAF000$_{H}$にdays1がアドレスB0000$_{H}$に確保されたと想定している

図3 文字列のメモリ表現

'H'はHのASCIIコードを意味する。'¥n,はC言語ではASCIIコード10_Hの文字として扱われる

char days1 [12]=
{31, 28, 31, 30,
……, 30, 31};

としてデータをセットすることができる。図2をご覧いただきたい。これは，仮にdays1が$B0000_H$に確保されたと想定した場合のメモリの様子である。days2のほうはあとで説明するので，ここでは無視しておいていただきたい。

そして，ここからが重要なのだが，配列のそれぞれのデータを扱う場合には，days1 [0]，days1 [1]，……，days1 [11] のようにして，データの塊の中の必要な部分だけを操作できるようになっている。char型のデータ12個分の領域を確保したので，データの最後はdays1 [11] になる点に注意されたい。

再び図2であるが，この図はひとつの重要な概念を諸兄に把握していただこうと用意したものである。さきほど「char型のデータ12個分のスペースを確保し，その先頭アドレスにdays1という名前をつける」と説明した。アドレスdays1にはchar型のデータが入っているので，days1は「charへのポインタ」と見なしても差し支えない。実際，＊days1で1月の日数を取り出すことができ，＊(days1＋1)で2月の日数を取り出すことができる。逆に，図2のようにdays2というcharへのポインタを入れる変数を用意し，days2にアドレスdays1をセットしてやれば，days2 [0]，days2 [1] などの表記でデータを取り出すことも可能になる。このように，配列とポインタは同じように扱うことができるのである。ただし，両者には決定的な違いがある。

図2をよくよく眺めていただきたい。days2はchar型のデータが入っているアドレスを収めるための変数である。アドレスを収めるのに必要なスペースが確保され，したがって，別のアドレスを，

days2=(char ＊)0xA0000;

のようにしてセットし直すことができる。ところがdays1はアドレス$B0000_H$につけられた名前にすぎない。

days1=(char ＊)0xC0000;

のように新しいアドレスをセットしようにも，この新しいアドレスを入れるためのスペースはどこにも用意されていないのである。ポインタと配列が同じように扱えるために，このことはしばしば忘れられがちである。

さまざまな配列

前回，C言語には文字列型というものはなく，文字のASCIIコードをメモリに並べ最後に0をセットしたものの先頭アドレスでこれを表現することを説明した。

printf("Hello¥n");

と書けば，メモリの適当な場所に図3のように文字列のASCIIコードが格納され，その先頭アドレス(図3では$B0000_H$)がprintf関数に渡されることになる。アドレス$B0000_H$にはchar型のデータが入っているのだから，これはchar型へのポインタである。実際，

char ＊str="Hello¥n";

のように，文字列の先頭アドレスをcharへのポインタ変数に格納することができる。

ここでポインタと配列の同等性を思い起こせば，文字列の先頭の文字は「＊str」とするだけでなく，「str [0]」としても取り出せることがおわかりであろう。文字列の2文字目は同様に「str [1]」とすれば取り出すことができる。また，文字列はcharへのポインタにすぎないことを考えれば，

str=str+1;

とすれば，先頭の1文字を除いた"ello¥n"という文字列とすることも可能である。

ここまで，char型の配列とcharへのポインタだけを対象に話をしてきたが，shortの配列やintの配列では面白いことが起きる。

int test [10];

と宣言すれば，testという記号は配列としてもポインタとしても使用可能となることは前述のとおりである。char型の配列では，＊(days1＋1)とすれば配列の次の要素を取り出すことができたが，int型の配列でも＊(test＋1)とすれば続く要素を取り出すことができる。すなわち，

test [n] ⟷ ＊(test＋n)

という対応づけがどのような型の配列であっても可能なのである。また，intへのポインタ変数testpを用意し，

testp=test;

としてこれにアドレスtestをセットしたあと，

testp=testp+1;

のように1つ大きくすると，testpはtest[1]のアドレスを示すように大きくされる。吾輩は，intを4バイトデータとして扱うので，testpは4バイト大きくなるわけである。

このように見てくると，ポインタと配列の同等性のため，メモリに入っているデータをどちらの形式で扱うべきなのか迷われるかもしれない。御仁はそもそも根が大雑把にできているので，これらを厳密に区別して使うことは少ないようである。吾輩が携わってきたこれまでのプログラムを見てみると，式の途中でいきなりポインタに［ ］をくっつけて配列的に処理したかと思えば，別の式では＊(配列名＋n) のようなポインタ的な使い方をしたりと，まさに混沌としている。

しかしながら御仁なりに基準というものはあるらしく，おおまかに次の点をよりどころに両者を使い分けていると考えられる。すなわち，処理の過程で注目する対象がある特定のアドレスに入っているデータだけである場合

にはポインタを，データ間の関係に注目する場合には配列を，という調子である。ポインタではあるが，明らかに配列的な使い方をするとわかっているものは，

char strp [ ]；

と宣言している。これは，

char ＊strp;

と宣言するのと同等であり，charへのポインタを用意しているのだが，配列的な使い方をするという御仁の意志の現れと見てよかろう。御仁の使用する変数はこの姿勢に従って宣言される傾向にある。にもかかわらず両者の使い方が混交するのは，面倒臭さを回避するに手段を選ばない御仁の悪習のなせる業である。吾輩はどちらの方法で書かれようと構わないのだが，この悪弊のいちばんの被害者は当の御仁自身であることに一向気づく様子はない。戯れに作ったプログラムとて，1月もたたぬうちに自分でさえ読みが下らぬようではなんのためのプログラミングか。諸兄はこのようなことのないよう注意されたい。

多次元の配列

人間というものは，なぜか表の形にデータを整理することが好きなようである。いや，好きというより表の形にせずにはいられないものらしい。吾輩なんぞは，メモリをざっと走査すればデータの概要を把握するのも朝飯前だが，人間はある特定の秩序をもってデータを並べてみなければ，そのデータが意味するものを把握するのもかなわぬということである。なんとも不憫な話ではないか。

コンピュータ言語というものも，直接マシン語を扱うのはかなわないということで考え出されたものであれば，表のように人間にとって扱いやすい形態も当然考慮に入れられている。図4をご覧いただきたい。これは小学校でおなじみの九九の表である。このような2次元の表を扱うのにC言語では，

char kuku [9] [9]；

のように表記する。これは，char型のデータが9個並んだ配列が，さらに9個並んで配列となっていることを意味している。

ここで問題となるのは，2つの[9]のうちどちらがX方向の順番を意味しているのかということだろう。変数の宣言は，

変数の型 変数名；

という形で行うことはご承知のとおりである。配列の場合も例外ではない。見やすさを考慮して要素数を変数名の後ろにつけ，

char test [9]；

のように表記はするが，これは「char型のデータが9個集まったもの」という型のtestという変数を用意しているにすぎない。char [9] というのが変数の型であり，test

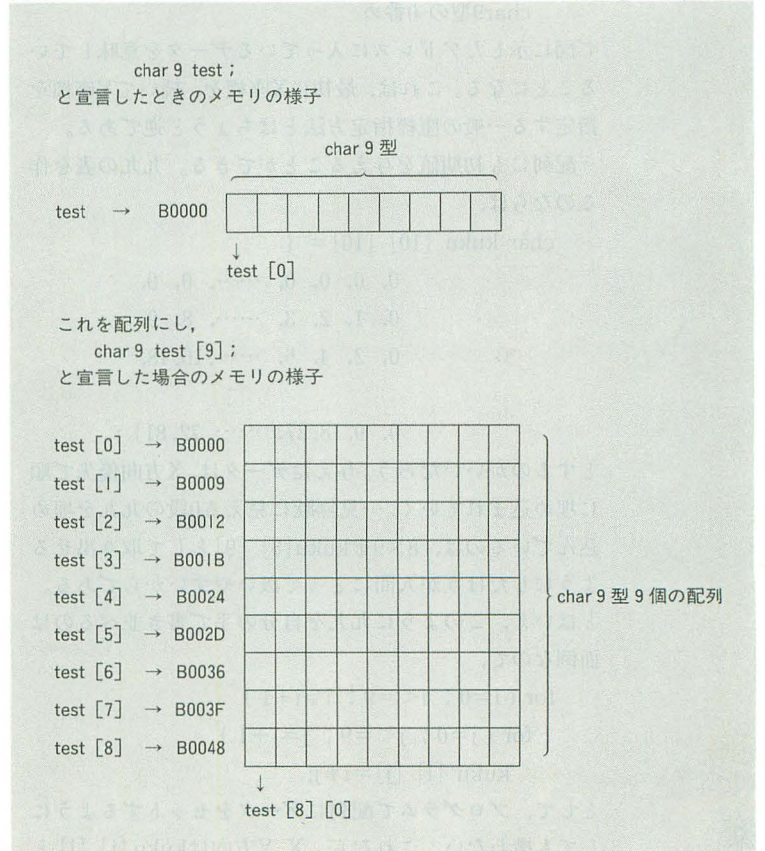
図4　昔懐かしの九九の表

	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81

図6　2次元配列とポインタ配列の違い

AF000	0xB0000
AF004	0xB0009
AF008	0xB0012
AF008	0xB001B
AF008	0xB0024
AF008	0xB002D
AF008	0xB0036
AF008	0xB003F
AF008	0xB0048

```
char *test [9];
testl [0]=test [0];
testl [l]=test [l];
……
testl [8]=test [8];
```

とした場合のメモリの様子。
testl[0][3]でtest[0][3]
testl[8][0]でtest[8][0]
を取り出すことができるが，実際のメモリの様子はまったく異なっている

図5　char 9 型の変数のメモリの様子

図7 argvの様子

void main (int argc, char＊argv [ ])
……
というmain関数をもつtest. xを，
A>test a b c
として実行した例。コマンドラインの文字列がメモリに格納され，その先頭アドレス（charへのポインタ）を収めた配列が作成され，そしてその先頭アドレスがポインタ変数argvへ与えられる

というのが変数名なのである。仮にこの型をchar9と名づけることにすれば，上の宣言は，

char9 test;

と宣言しているのと同じことになる。すなわち，図5のよう，79バイトの大きさをもった領域が，変数testとして割り当てられると考えていい。一般の変数と異なるのは，C言語は配列をひと塊のデータとして扱うことができないので，配列名だけを書くと配列の先頭アドレスを示すようになっている点である。

このchar9型データの配列も図5に示してある。test［0］～test［8］の9個のchar9型データが並んで配列となっているのがおわかりいただけるだろう。test［8］はchar9型のデータである。したがって，

```
test [8]   [0]
└──────┘   └──┘
char9型の  0番め
```

で図に示したアドレスに入っているデータを意味していることになる。これは，最初にX座標を，続いてY座標を指定する一般の座標指定方法とはちょうど逆である。

配列にも初期値を与えることができる。九九の表を作るのならば，

```
char kuku [10] [10]= {
        0, 0, 0, 0, ……, 0, 0,
        0, 1, 2, 3, ……, 8, 9,
        0, 2, 4, 6, ……,16,18,
                   ⋮
        0, 9,18,27, ……,72,81};
```

とするのがいいだろう。与えたデータは，X方向優先で順に埋め込まれていく。一見無駄に見える0段の九九を埋め込んでいるのは，8×9をkuku[8]［9］として取り出せるようにしたほうが人間にとって扱いやすいからである。とはいえ，このように九九を自分の手で書き並べるのは面倒なので，

```
for ( i=0; i<=9; i=i+1 )
  for ( j=0; j<=9; j=j+1 )
    kuku [i] [j]=i＊j;
```

として，プログラムで配列にデータをセットするようにしても構わない。これなら，X-Y方向はkuku［i］［j］と書くか，kuku［j］［i］と書くかによって，自分の好きなように設定することができる。掛け算は可換なので，いずれの順でセットしても出来上がる表に変わりはないが。

図5を見てお気付きの諸兄がいらっしゃるかと思うが，2次元配列も1次元配列もメモリの様子は変わらない。C言語では配列の添え字が宣言された範囲にあるかどうかはチェックされないので，test［0］［72］とtest［7］［9］とtest［8］［0］は同じメモリを意味する。

char9型のデータ，test［0］～test［8］は配列なので，配列名がその先頭アドレスを意味するのと同じく，$B0000_H$，$B0009_H$，……，$B0048_H$を意味することになる。これらのアドレスにはchar型のデータが入っているので，test［0］～test［8］の型はいずれもcharへのポインタとみなして差し支えない。とはいうものの，test［0］～test［8］をcharへのポインタを収めた配列と混同しないよう注意されたい。こちらは，

char ＊test1［9］;

のように宣言され，図6のようにメモリに格納される。2次元配列との違いがおわかりだろうか。

配列とメモリの深淵へ

2次元配列も配列であるから，testやkukuのような配列名は配列の先頭アドレスを示している。このアドレスにはchar型のデータが入っているので，testやkukuもcharへのポインタであると思われるかもしれない。しかしながらこれは間違っている。なぜなら，test［0］として取り出すことができるのはcharではなくchar9型のデータのアドレスだからである。したがってtestやkukuをポインタ変数に格納するには，

char (＊testptr)［9］;

として，char9型のデータへのポインタ変数を用意する必要がある。＊testptrにカッコがつけてあるのは，これがないと演算子の優先順位により，

```
char ＊ testptr  [9];
└──────┘         └───┘
charへのポインタの 配列
```

と優先順位がつけられてしまうからである。

とはいってもC言語はポインタの代入には寛大に作られているので，

```
char ＊testptr;
testptr=test;
```

のようにしても，警告が発せられるだけでエラーにはならない。吾輩の純正CコンパイラであるXCでは警告すら発せられない。testptrをこのように宣言するなら，直観どおりtestptr［0］で，アドレスtestに入っているchar型のデータを取り出すことができるようになる。

さて図7は，

A>test a b c

のようにパラメータを指定して起動されるプログラムを作成するときに使う方法である。

```
void main( int argc,  char *argv [  ] )
```

のようにmain関数のパラメータとして，argc，argvの2つを指定すればこのようなプログラムを作成することができる。

main関数が実行される時点で，argcにはいくつのパラメータが与えられたか（コマンド名だけだとargcは1になる）が格納される。それぞれのパラメータ文字列はメモリに収められ，そのアドレスが配列として並べられ，そして，argvにはその配列のあるアドレスが与えられる。したがって，argvはchar(パラメータ文字列)へのポインタが収められたメモリアドレス(ポインタ)を保持することになる。変数の型でいえば，

```
char **argv;
```

ということになるが，配列的に使うことを明示するためこのように，

```
char *argv [  ];
```

と書くのが習慣となっている。

ここまで読み進まれた諸兄には，実際に与えられたパラメータの最初の文字を取り出すにはどうすればいいかもうおわかりのことと思う。argv [0] とすればコマンド名が入っているアドレスが取り出せる。そのアドレスを仮にargv0とすれば，アドレスargv0に入っている文字はargv0 [0] とすれば取り出すことができる。2文字めはargv0 [1] である。続けて書けば，argv [0] [0] でコマンド名の1文字めが，argv [0] [1] で2文字めが取り出せるわけである。コマンドの中にはスイッチと呼ばれるパラメータをとるものがある。スイッチは“—”あるいは“/”で始まる文字列で，コマンドの機能を拡張したり制限するのに使用される。たとえば，dirコマンドを，

```
A>dir b: /p
```

のように使えば，ファイル名を1ページ表示したところで自動的にキー入力を求めるようになる。このようなスイッチを判定するのに，argv [n] [0] は有効である。

メモリ自由自在

若手の台頭で気をはくヤクルトの健闘によって，今年のペナントレースは例年になく面白くなっていると御仁はニュースを楽しみにしている。阪神にしか勝ち越していなかった巨人もようやく大洋に勝ち越し，今後の動向も気をひくところらしい。

吾輩はメモリを自由に扱うことができる。したがって，吾輩が野球の勝敗表をメモリに収めるとしたら，おおむね図8のような形にするであろう。これは7月上旬のセ・リーグの勝敗表である。チーム名は，チーム名のASCIIコードを収めたメモリの先頭アドレスで表現し，続けて試合数や勝敗を並べるのである。このような表をC言語の配列を使って表現することはできない。配列は同じ型のデータを表の形にするものだからである。

C言語は型の違うデータをひとつにパックするために構造体というデータ型を用意している。構造体を使えば，図8のB0000$_H$～B0007$_H$に収めた1チーム分の勝敗は，

```
struct table  {
    char  *team;
    char games;
    char wins;
    char losts;
    char draws;
};
```

と表現される。これは「struct table型」という型を新たに定義しているところである。以後，

```
struct table july;
```

のように宣言すれば，struct table型の変数julyを用意することができる。もしこの構造体を一度しか宣言しないならば，

図8 構造体とメモリ

1.セ・リーグ勝敗表

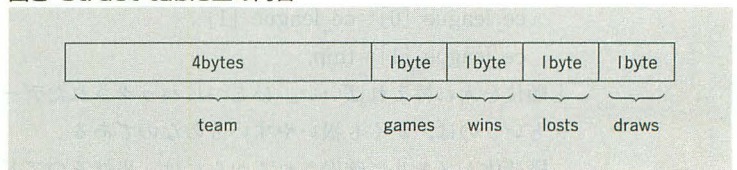
	試合	勝	敗	分
中日	65	40	24	1
ヤクルト	70	40	29	1
巨人	71	38	33	0
広島	60	31	28	1
大洋	69	30	38	1
阪神	67	20	47	0

2.勝敗表のメモリ表現

AF000	'中'	'日'	0		
AF005	'ヤ'	'ク'	'ル'	'ト'	0
AF00E	'巨'	'人'	0		
AF013	'広'	'島'	0		
AF018	'大'	'洋'	0		
AF01D	'阪'	'神'	0		

B0000	0xAF000	65	40	24	1
B0008	0xAF005	70	40	29	1
B0010	0xAF00E	71	38	33	0
B0018	0xAF013	60	31	28	1
B0020	0xAF018	69	30	38	1
B0028	0xAF01D	67	20	47	0

図9 struct table型の内容

4bytes	1byte	1byte	1byte	1byte
team	games	wins	losts	draws

```
struct {
   char *team;
   char games;
   char wins;
   char losts;
   char draws;
} july;
```

と一気に宣言して構わない。structと{の間に名前(タグ名)を書くのは,何度か繰り返して宣言するときの簡略化のためである。

この変数julyは図9のように合計8バイトからなる変数である。構造体の宣言のときに用いたteam, gamesなどの名前は構造体の構成要素につけた名前で,構造体のメンバと呼ばれている。これは変数julyの中のどの部分を扱うかを指示するのに使用する。julyにチーム名をセットするなら,

```
july.team="中日";
```

julyに試合数をセットするなら

```
july.matches=64;
```

のように,「変数名+ピリオド+メンバ」で操作する部位を指定するのである。

構造体は複数のデータをパックするものだと説明した。これは,

```
struct table tmp;
tmp=july;
```

と同じ型の構造体を宣言すれば,普通の変数のように自由に代入できるためである。また,

```
struct table shohai( void )
        :
```

と,struct tableを返す関数を作成することもできる。

図8でメモリに作成した勝敗表は,このstruct table型を配列にし,

```
struct table ce_league [6]={
  "中日", 64, 40, 24, 1,
  "ヤクルト", 69, 40, 29, 1,
                :
}
```

とすれば作成できる。ここでは初期値代入を使って一気に表を作っているが,

```
ce_league [0].game=64;
```

のように1つひとつセットしていっても構わない。今後試合が進んでいってヤクルトが中日を逆転すれば,

```
struct table tmp;
tmp=ce_league [0];
ce_league [0]=ce_league [1];
ce_league [1]=tmp;
```

と順位を入れ替えればいい。ひとつにパックされたデータというのは,かくも扱いやすいものなのである。

構造体もメモリに確保されるからには,当然そのアドレスというものを持っている。これまでの宣言と同じように,構造体のアドレスを保持するポインタ変数は,

```
struct table *ptr;
```

として宣言できる。

```
ptr=&ce_league [1];
```

のように構造体のアドレスをセットすると,*ptrで構造体そのものを,(*ptr).teamでチーム名を取り出すことができる。ポインタ変数からメンバを操作するのに,このように書くのは面倒だということで,ptr―>teamという表記法も用意されている。プログラム作成時のタイプ数の少なさが自慢のC言語ならではというところか。

概括の終わりに

最後にC言語のプログラムでよく見かける表記について触れておこう。

```
typedef unsigned short UWORD;
```

は,UWORDという新しい型名を定義する。これは,

```
typedef 既存の型名 新しい型名
```

という文法で使用され,長々しい型名を書くのが面倒なときに重用される表記である。

この定義方法は,一般の変数宣言の変数名を書くところに新しい型名を書き,行頭にtypedefを書き加えると覚えるといいだろう。したがって,九九の表で使ったchar9型という型は,

```
typedef char char9 [9];
```

とすれば定義することができる。以後,

```
char9 test;
```

のように書けば,それは,

```
char test [9];
```

と宣言しているのと同じことになる。また,

```
typedef struct {
  char *team;
  ……
  char *draws;
} table;
```

と書けば,勝敗の構造体にtable型という名前をつけることができる。勝敗表の宣言は,

```
table ce_league [6];
```

のようになる。

3回にわたってC言語の概要をお届けしてきた。どのようなコンピュータ言語であっても,メモリを使って表現できない変数を扱うことはできない。C言語の難関とされるポインタにしても,メモリに収められた変数とそのアドレスという見方をすれば,じつに簡単なものだということがおわかりいただけるかと思う。複雑な構造体が宣言されている場合は,そのメモリイメージを描いてみるといいだろう。図で眺めるほうがはるかに人間には理解しやすいものなのだから。

ポインタって何だろう（前編）

［第11回］

Nakamori Akira
中森 章

さて，今月はいよいよポインタのお話です。C言語を学ぶにあたって，このポインタでつまずくといった人が多いといわれますが，今回の中森氏の解説を読めば，実はそう難しいものではないと感じるはずです。がんばって制覇してください。

ついにXVIを買いました（ハードディスク付き，メモリは8M実装！）。フロッピーディスクを使って初代X68000のハードディスクからXVIのハードディスクへファイル転送をしたのですが，フォーマットやディスクコピーのスピードの歴然とした差に，クロック1.6倍を実感した中森章です。これで，原稿を書くスピードが1.6倍になってくれればいいのだけれど……。

さて，今回のテーマはお待ちかねのポインタです。ポインタはC言語で最も特徴的なデータ型です。ポインタというデータ型はPASCALにもありましたが，それは抽象的で非常に難しい概念でした。しかし，C言語のポインタは単純明快です。よく初心者はこのポインタという壁に突き当たってC言語に挫折してしまうといわれますが，冷静な目で眺めれば大したものではありません。今回はこのポインタの基礎事項について学んでいくことにしましょう。

ポインタを学ぶ上での基礎知識

C言語を学ぶ人にとってポインタという概念が非常にとっつきにくいものであることは，ポインタだけにトピックスをしぼった参考書（参考文献1)2)など）がいくつか出版されていることからもわかります。しかし，ポインタとはそこまで構えて学ぶようなものでしょうか。

かつて祝一平氏は，
「ポインタがわからないことは（HL）や（A0）がわからないということであり，ひいては，コンピュータがわかっていないことなのだ」
といっています（参考文献3)）。これは結構的を射た発言です。（HL）や（A0）はアセンブリ言語でメモリの内容を参照するときの基本的なアドレッシングモードのひとつです[1]。祝氏のいうとおり，このアドレッシングモードが使いこなせないようでは，**アセンブリ言語**でまともなプログラムを書くことはできないでしょう。逆にいえば，アセンブリ言語を知っている人にとっては，ポインタという概念は「（HL）とか（A0）といったアドレッシングを高級言語風に実現しただけ」というひと言で済んでしまう程度のものなのです。もともとC言語はシステムを記述するための高級なアセンブラとして開発されてきました。ポインタはアセンブリ言語で普通に書いていた処理を高級言語でも書くことができるように取り入れられた概念なのです。

ポインタとはアセンブリ言語の概念です[2]。このため，ポインタを理解するためには最低限のアセンブリ言語の知識が必要になります。この連載はC言語を高級言語的な立場から見て解説するのが目的でしたが，ポインタを学ぶうえではアセンブリ言語という関所を避けて通ることはできません。ただし，ここで必要な知識とはそれほど難しいものではありません。むしろ，アセンブリ言語や高級言語にかかわらず，プログラムを書く人すべてが知っておくべきコンピュータの常識なのです。それは次の3つの事項です。

- **プログラムやデータはメモリに格納される**
- **メモリには連続的な番号（アドレス）が付けられていて，メモリの内容はその番号で参照される**
- **メモリのひとつのアドレスに対しては1バイトの大きさの領域が割り当てられている**

簡単にいえば，メモリ空間はアドレスが添字となるchar型の配列と見なせるということですが，わかりますよね。もし，この3つの事項のうちどれかひとつでもわからないことがあったら，コンピュータに精通した友人を探して，教えてもらってください。これから先の説明はこの3つの事項を理解しているものとして進めていきます。また，これら3つの事項が理解できないようなら，残念ですがポインタをマスターすることはできないと思ってください。

さて，上の事項はコンピュータに関する常識でしたが，ポインタを理解するためにはさらにプログラミング言語についての常識も必要です。これも覚えておきましょう。それは次の3つの事項です。

- **変数や配列の内容を格納する領域はメモリに割り当てられる**
- **変数や配列はデータ型に応じたバイト数の連続領域（アドレスが連続する）をメモリ上で占める**
- **変数名や配列名はそれが占めるメモリ領域の先頭アドレスを示す定数である**

これらは先に示したコンピュータに関する3つの常識の上に成り立つものであるということはいうまでもありませんね。

ただ，3番目の事項は少しわかりにくいかもしれません。これは，たとえば，“変数x”が“アドレス1000番地”と同じものということです。つまり，1000番地に変数xが割り当てられたら（1000番地から変数xの内容が格納される），そこが最後まで変数xの内容の格納場所になるということです。通常，変数xの内容の格納場所がほかのアドレスに移ることはありません[3)]。“変数xの内容は……”という表現と“アドレス1000番地の内容は……”という表現は同じことになります。プログラムで“変数x”と書かれている部分は，実際には“アドレス1000番地”と読み変えられて処理されると思ってもよいでしょう。

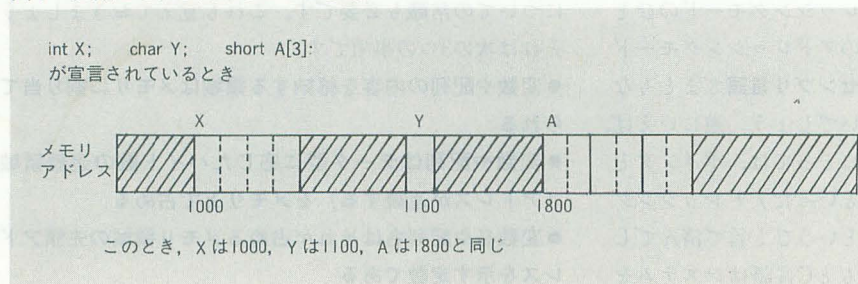
とにかく，この3つの事項は非常に重要ですから，図1に変数とそれが占めるメモリ領域の状態概念図を示しておきます。このイメージをしっかりと頭に叩き込んでください。

以上の6点の事項を押さえたところで，いよいよポインタの説明に入ることにしましょう。

1）(HL)はZ80，(A0)はMC68000のアドレッシングモードで，どちらもレジスタ間接を示す。アドレッシングモードとはCPUがメモリを参照するときの方式である。この場合，参照するメモリのアドレスがHLまたはA0というレジスタに入っている。
2） PASCALでいうポインタはアセンブリ言語の概念とは少しかけ離れている。C言語のポインタには「メモリ領域」という具体的なイメージがつきまとうが，PASCALのポインタはもっと抽象的である（だから理解しにくいかもしれない）。
3） 反例をあげればきりがない。ここでは，それらを指摘して話をややこしくすることはしない。

ポインタとは何か

ポインタとはデータをポイント（指し示す）するものという意味です。ここでいうデータとは変数や配列のことです。C言語では変数や配列を指し示すための変数が用意されていて，それをポインタと呼んでいます。ポインタとは変数や配列をポインタという変数を通して間接的に参照するための仕組みなのです。どんな処理のためにポインタが必要になるかはあとで説明することにして，ここではポインタというものの具体的なイメージをつかむことにしましょう。

図1 変数とそれが占めるメモリ領域

すでに述べたように，変数や配列はメモリ（レジスタの場合はここでは考えない）上にその領域が確保されています。つまり，変数や配列の占める領域はその名前で特定することができます。変数や配列の名前はメモリのアドレスと同一です。すなわち，ポインタの指すものはメモリのアドレスということになります。結局，ポインタとはメモリのアドレスを保持している変数にすぎません。これがポインタのすべてです。

さて，ポインタが保持するものはメモリのアドレスですが，そのアドレスはポインタが指し示す変数や配列のデータ型と密接な関係を持っています。次のような表現を考えましょう。

1） 1000番地にはchar型データが格納されている
2） 1000番地にはint型データが格納されている
3） 1000番地にはdouble型データが格納されている

これらはどれもよく使われる表現ですが，厳密には2)と3)の表現は正しくありません。ひとつのアドレスで参照できるのはメモリの1バイトの領域だけです。すなわち，2)，3)は，

2） 1000番地から始まる4バイトの領域にはint型データが格納されている

3） 1000番地から始まる8バイトの領域にはdouble型データが格納されている

が正しい表現です。いま，ポインタに1000という値（アドレス）が入っていたとしましょう。このとき，1000というのはメモリ上の1点（1バイト）を示すアドレスです。しかし，ポインタとして1000番地を参照するときは，1000番地から格納されているデータのデータ型によって意味が微妙に違ってきます。つまり同じ1000番地でも，

1） char型を格納した1000番地
2） int型を格納した1000番地
3） double型を格納した1000番地

というように異なる意味を持つことになります。当然，ポインタ（1000番地）の指し示すデータを参照するときは，そこにある1バイトのデータだけを参照したのでは意味がありません。ポインタがchar型データを指し示す場合はそれでいいのですが，int型データを指し示す場合は1000～1003番地に格納された4バイトを，double型データを指し示す場合は1000～1007番地に格納された8バイトを参照しなければなりません。このように，ポインタはメモリ上の1点を指し示すアドレスを保持するだけですが，それが指し示すデータ型と対になっていなければ使いものにならないのです。ポインタというものにはデータ型が付属しているということを覚えておきましょう。

ポインタの宣言と参照方法

それでは，C言語でのポインタの説明に移ることにしましょう。ポインタといえども変数には変わりありませんから，そこで必要なのはほかの変数と同様に変数の宣言と値の参照方法です。それらを説明しましょう。

●ポインタの宣言

ポインタの宣言は普通の変数の宣言とまったく同じ形式で行います。ただ違うのは変数名の前に＊という記号をつけるということだけです。たとえば，

long int ＊someWhere;

という宣言はlong int型のデータが格納されているアドレスを保持することのできるsomeWhereというポインタを宣言することになります。＊を「指し示す先」と読み換えて，指し示す先がlong intであるような変数(これがポインタ)someWhereを宣言していると思えばよいでしょう。また，

char ＊string［100］;

は配列stringの各要素の指し示す先がchar型データであるような配列を宣言していることになります。これは各要素がchar型データを指し示すポインタであるような配列の宣言です。すなわち，string［1］とかstring［10］とかいったそれぞれの要素がchar型データを指し示すポインタとなるのです。

ポインタを宣言するときの＊はひとつとは限りません。

float ＊＊noWhere;

というような宣言も可能です。これは，

float ＊(＊noWhere);

と，読み換えてポインタnoWhereの指し示す先（＊noWhere)がさらにfloat型データを指し示すポインタとなっていることを意味しているのです。このように考えれば，

double ＊＊＊＊＊＊＊＊indirects;

というポインタの宣言も理解できるでしょう（これを扱うのはプログラマの能力を超えるかもしれないが)。

●ポインタが指し示すデータの参照

ポインタが指し示すデータを参照するには，ポインタの宣言のときに使った＊を演算子として使用します。＊は間接演算子と呼ばれる演算子です。この演算子をポインタの前につけるとポインタが指し示すデータを参照できるようになります。

long int ＊someWhere;

というような宣言がなされているとき，

a = ＊someWhere + 100;

という式はsomeWhereの指し示すlong int型のデータに100を加えてaに代入するという意味になります。また，

＊someWhere = 1234;

のように＊をつけたポインタが=の左辺にあるときは，1234という整数をsomeWhereの指し示す場所(メモリ上の連続するどこか4バイト）に格納することを意味します。この＊という記号はポインタの宣言のときにつけたのと同じだけの個数をポインタにつけることによって最終的な変数や配列の値に到達することができます。ポインタに宣言のときよりも少ない個数の＊をつけて参照できるのは，次に＊で参照すべきアドレスを保持しているポインタなのです。

int ＊＊＊a;

という宣言がされているとき，

a

＊a

＊＊a

＊＊＊a

という形式で参照を行ったとき，それぞれの関係がどうなるかを図2に示しておきます。いまの場合，＊を4個つけた＊＊＊＊aには意味がありません。これではint型データの値の指し示す先（なんだそれは）を参照しようとしてしまいます。このようなときCコンパイラはエラーメッセージを出すでしょう。

実はポインタの指し示す先のデータを取り出す演算子は＊だけではありません。配列の要素を参照するときに使用した［ ］という演算子を使用することもできます。配列要素は配列名に［ ］という演算子をつけて参照しますが，配列名は配列の先頭アドレスを示す定数値です。つまり，［ ］はアドレスに対して適用することのできる演算子なのです。アドレスとくると連想的にポインタという言葉が思い浮かんできます。配列名はそのものがアドレス，ポインタはそれが保持するデータがアドレス，という違いがあります（ポインタの場合はメモリからアドレスを取り出さないと参照できない)が，［ ］という演算子はポインタに対してもちゃんと適用できるようになっているのです。＊と［ ］の演算子の関係についてはあとで説明することにして，とりあえず，ここでは［ ］でポインタが指し示すデータを参照する方法を示しましょう。それは，ポインタに対して［0］を適用することです。つまり，

int ＊noWhere;

が宣言されているとき，

noWhere［0］

は，

＊noWhere

とまったく同じデータを参照します。

●ポインタの初期化

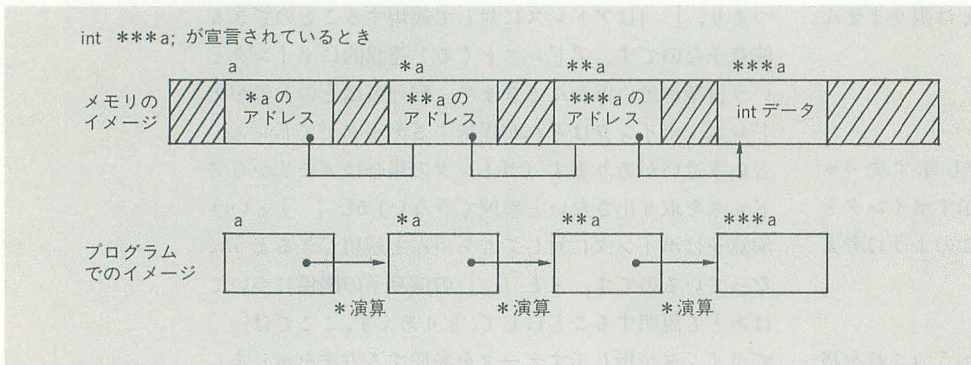
ポインタは変数ですから，通常の変数と同じく，宣言しただけでは有効な値を持っていません[4]。このため，おそらく，

int ＊noWhere;

などというポインタを宣言した直後にいきなり，

＊noWhere＝1234；
などという式を実行するとバスエラーが起きてしまうでしょう。これはnoWhereの保持するアドレスが0またはとんでもない値を示しているため，そのアドレスへの書き込みが保護違反になってしまうためです。このようなポインタ参照を行った場合，X68000のようにシステムエリアとユーザーエリアが区別されていて，アプリケーションからシステムエリアを保護しているシステムならバスエラーだけで済んでしまいますが，保護機構のないシステムでは簡単に暴走してしまうでしょう。実際，ポインタに関するトラブルのほとんどは初期化を忘れたことによるものだといわれており，このようなポインタはワイルド（荒々しい）ポインタと呼ばれプログラマに恐れられています。

ポインタは変数や配列のアドレス，あるいはほかのポインタのアドレスを保持する変数ですが，ポインタを使用するためには前もって値（アドレス）を与えてやらなければなりません。そのためにC言語では変数のアドレスを取り出す&演算子が用意されています。変数や配列要素の前に&をつけるとその変数や配列要素のアドレスを取り出すことができます。たとえば，上で宣言したnoWhereというポインタをint型変数xのアドレスで初期化するためには，

　noWhere = &x;

という式を実行すればよいのです。また，int型データを要素とする1次元配列Arrayの10番目の要素のアドレスでnoWhereを初期化するためには，

　noWhere = &Array [9]；

という式を実行することになります。どちらの場合も通常に使用する変数や配列要素にそのまま&をつけてアドレスを取り出しているだけです。このときはポインタに値を与えるのであって，その指し示す領域に値を与えているのではないので＊はついていません。なお，お気づきのように，

　＊&x

という式は，xという変数のアドレスが指し示すものという意味ですから，xそのものと同じになります。また，register宣言した変数にはアドレスというものがありませんから，それに&演算子をつけることはできません[5]。

実際のC言語のプログラムでは，ポインタの初期値として配列の先頭アドレスを与えることがよくあります。これまでの説明でわかるように，配列の先頭アドレスは（1次元配列の場合は），

　&Array [0]

などと配列の1番目の要素に&をつけて取り出すことができますが，C言語では配列名そのものがその配列の先頭アドレスを示すことになっています。

そこで，

　int Array [100]；
　int ＊noWhere = Array;

というような初期化をすることもできます。また，配列名と変数名はメモリのアドレスのある位置を示しているという意味で同等のものです。変数xのアドレスを取り出すための式が，

　&x

と表現できるのなら，配列Arrayの先頭アドレスを取り出すための式が，

　&Array

と表現できてもいいと思いませんか。この表現はプログラムであまり見かけたことはありませんが，これもANSI Cでは実際に許されている表現です（ANSIに準拠していないCコンパイラでもエラーになるものはないようだ）。結局配列の先頭アドレスを示す表現は，

　Array
　&Array
　&Array [0]

の3通りがあるということです。配列の先頭アドレスを取り出すことは頻

図2　ポインタの指し示すもの

図3　ポインタの加減算

繁に行うのでこれらの表現はぜひ覚えておきましょう。

さらに，よく見掛けるポインタの初期化としては文字列による初期化です。文字列とは，いうまでもなく，”と”で文字を囲んだあれです。この文字列はそれ自体がアドレスとなることができます。以前説明したように，C言語では文字列とは要素が文字列の各1文字であり，最後の要素がNULL (0) であるchar型配列として扱われます。すなわち，

```
"This is a pen"
```

という文字列は，

```
char str []= {
    'T', 'h', 'i', 's', ' ', 'i', 's',
    ' ', 'a', ' ', 'p', 'e', 'n', 0
};
```

と宣言されたchar型配列と同じものとして扱われます。後者はstrという名前を持っていますが，前者には名前がなく”This is a pen”という表現で直接配列の先頭アドレスを示します。したがって，文字列はポインタ変数にそのまま代入できるのです。たとえば，

```
char *str1="This is a pen";
```

という具合です。この場合，“This is a pen”という文字列を構成する文字はchar型の配列としてメモリのどこかに確保されて，その先頭アドレスがstr1というポインタの初期値となります。ポインタを文字列で初期化するとき，文字列の要素はchar型ですから，それをchar型とは異なるデータ型（たとえばint型やdouble型）のポインタ変数に代入すると，もはや元の文字列とは違った意味になってしまいますから注意しましょう。なお，ポインタとは直接関係ありませんが，

```
char str2 []="This is a pen";
```

という宣言は，上のchar型配列strの宣言とまったく同じものです。char型データへのポインタであるstr1との違いに注意しましょう。str1は”This is a pen”という配列の先頭アドレスを保持している**変数**，str2は”This is a pen”という配列の先頭アドレスを示す**定数**です。

ところで，アドレスというものは（符号なしの）整数値と同一視できます（これに疑問を持つ人はいますか？）。したがって，ポインタ変数にアドレス値を絶対的な整数値で与えたければ，

```
short *GRAM = 0xc00000 ;
```

などという方法も考えられます。これで，プログラムからメモリの内容を絶対的なアドレスを指定して参照することができるようになります。しかし，ポインタに与えるアドレスがプログラムで使っていい領域かどうかわからない（もしかしたらOSが使うワークエリアかもしれない）ならば，こんな危険な真似はよしたほうがいいでしょう。上の例の0xc00000という値（アドレス）はX68000のグラフィックRAMの先頭アドレスですから，それを知ってプログラムを書くのであればかまいません[6]。

このほかに，ポインタ変数に値を与える方法としては別のポインタが保持しているアドレスをポインタの初期値として代入することも考えられます。何々型へのポインタ変数といってもその値はひとつのアドレス値でしかないわけですから，ポインタ相互間の代入はできて当たり前でしょう。あとで説明してあるポインタのキャストの項も読んでください。

---

4)　C言語では外部変数と静的変数は初期化の指定がない場合は0に初期化される。自動変数は初期化の指定がない場合には初期値として不定な値を持つ。
5)　GCCは自動変数をできるだけレジスタに割り付けようとするから，変数のregister宣言をしてもしなくてもほとんど同じコードが生成される。ただし，変数に＆がついた式があると，その変数だけはレジスタに割り付けない。なんと賢いコンパイラ。
6)　グラフィックRAMはシステムエリアに属するので，その領域をそのままプログラムから参照することはできない。バスエラーが発生する。その前に，ライブラリ関数を使ってスーパーバイザモードに移らなければならない。

ポインタの必要性

ここまで読んでこられた方はそろそろポインタがいったい何の役に立つのか知りたくなっているところでしょう。それでは，ここらへんでポインタの効能について考えてみましょう。K&Rを見るとポインタの必要な理由として次のような項目が挙げられています。

- **ときとして計算を表現する唯一の方法である**
- **ほかの方法で得られるよりもずっとコンパクトで効率的なプログラムが書ける**

第1の理由は，ポインタを使わなければできないようなプログラムがあることを示していると考えられます。K&Rの中に，具体的にこれといった実例が示されているわけではありませんが，これは関数の参照呼び出し(Call by Reference) のことをいっているのだと推測されます。

C言語の関数呼び出しは一般に値呼び出し (Call by Value)と呼ばれているものです。これは変数の値を引数という別の領域にコピーして関数に渡す方法です。関数が受け取るのは引数にコピーされた変数の値ですから，それを読むことはできても，その値を保持していた元の変数の値を変更することはできません（引数の値は変更できるがもとの変数の値には反映されない)。これを説明する例としてよく取り上げられるのが，2つの変数の保持するデータを交換するswapという関数です。これは次のような関数です。

```
swap(x, y)
int x, y;
{
    int work;
    work=x;
    x=y;
    y=work;
```

```
}
```

そして，この関数を呼び出す側が，

```
main()
{
        int a=2;
        int b=3;
        swap(a, b);
        printf("a=%d¥n", a);
        printf("b=%d¥n", b);
}
```

とするとき，変数aとbの値はうまく入れ替わってくれるでしょうか。答えはNOですね。関数swapは，変数aの値が引数xに，変数bの値が引数yにコピーされて呼び出されます。関数swapの中では期待通り引数xと引数yの値は入れ替わっています。しかし，引数であるxやyの値を変更したからといって，それに値をコピーしたaやbという変数の値が変更されることはありません。呼ばれた関数の側では，引数に渡される前の変数aやbの値が何であったかを引数xやyの値として知ることができるだけなのです。せっかく値を入れ替えたはずの引数xとyの値は関数swapを抜け出すときに捨てられてしまうのです。これが値呼び出しというものなのです。

この方法は引数として渡す変数の内容が呼んだ関数の中で勝手に変わってしまうことがないので，プログラムの流れを追いやすくなります。比較的保守のしやすいプログラムを書くことができるのです。その反面，上のswap関数のようなプログラムが書けなくなってしまいます。その問題を解決するための手法がポインタなのです。swap関数の引数をポインタで宣言しておき，その引数へ変数aとbのアドレスの値を渡すことにすれば，そのポインタ（変数のアドレス）に対して間接演算子（＊）を適用することで元の変数（アドレスが指し示すメモリ上の領域）の値を変更することができます。もっとも，これも一種の（アドレスという値による）値呼び出しですが，呼ばれる関数側の対処によって参照呼び出しと同じ効果を発生させているのです。具体的なswap関数は次のようになります。

```
swap(x, y)
int *x, *y;
{
        int work;
        work=*x;
        *x=*y;
        *y=work;
}
```

このとき，swap関数は，

```
swap(&a, &b);
```

によって呼ぶようにします。結局，参照呼び出し（的な機能）を実現することで値呼び出しが持っていたプログラムの読みやすさという特徴を失うことになりますが，背に腹は替えられないというところでしょうか。

また，この関数の参照呼び出し（的な機能）は関数から複数の戻り値を受け取るときにも役立ちます。一般に関数というものは戻り値をひとつしか持ちません。しかし，たとえば，マウスを扱うプログラムを書いていて，マウスカーソルの座標を返す関数が欲しくなったとします。このとき戻り値はX座標とY座標の2つが必要です。このように，関数からの戻り値を複数個持たせたいときがよくあります。ポインタが威力を発揮するのはこんなときです。関数というものは戻り値はひとつですが引数は複数個取ることができます。そこで，引数として戻り値を格納しようと思っている変数のアドレスを渡すのです。呼ばれた関数の側ではその引数をポインタとして受け取ります。そうすることによってポインタで指し示されるデータの値を変更することができるようになります。たとえば，マウスカーソルの座標を戻り値とする関数をmsposとして，その関数からマウスの座標を受け取る場合を考えます。まず，戻り値を格納するための変数x，yを宣言しておいて，

```
mspos(&x, &y);
```

という具合にmspos関数を呼び出せばよいのです。mspos関数の側では，

```
mspos(x, y)
int *x, *y;
{………}
```

と宣言して値を返すべき変数のアドレスをポインタ変数として引き取ります。この関数内ではマウスカーソルの座標を計算した（これはたとえの話ですからどうしたら座標を計算できるかまでは考えていません）あと，それぞれのポインタ変数の指し示す位置に，

```
*x=マウスカーソルのX座標;
*y=マウスカーソルのY座標;
```

として座標の値を入れればよいのです。これは，先ほどのswap関数とほとんど同じ発想です。

しかし，決定的な違いは引数としてアドレスを渡す変数の初期化にあります。つまり，swap関数は引数にアドレスを渡す変数に有効な値が入っているときに呼び出します（その変数の値を交換するのですね）。一方，mspos関数はそれ自身が引数として渡されたアドレスの指し示す先に値を入れてくるため，引数としてアドレスを渡す変数が初期化されていることはありません。これは少し問題です。私たちがプログラムを読むとき，変数は必ず初期化して使うものと思っていますから，ますます読みにくいプログラムになってしまうことになります。しかし，この欠点があるとしても，複数の戻り値を返す関数というのは非常な魅力なのです[7]。

第1の理由が長くなりましたが，ポインタが必要な第2の理由はプログラムが効率的に書けるということだそう

です。たぶん，配列で行っている処理をポインタで行うと実行速度が速くなるといった類のことをいっているのでしょう。これについてはあとで説明します。もっとも，最近のコンパイラは賢くなってきているので，ある処理を配列を使って書こうが，ポインタを使って書こうが実行速度に決定的な差が出るとは思えませんが。

7) 関数から複数の戻り値を戻す方法として構造体を利用する方法もある(構造体についてはまだ説明をしていない)。構造体のほうが扱いにくいのか，このような場合は必ずといっていいほどポインタが利用される。変数のアドレスを関数に渡して内容を書き換えてもらう例はSX-WINDOWのライブラリ関数によく見られる。

ポインタに関する演算

先にも述べましたように，ポインタは符号なし整数を保持する変数と同一視することができます。このため，ポインタに対して整数と同じ加減乗除および比較の演算ができると思いがちですが，世の中そんなに甘くはありません。C言語でポインタに対して許されている演算は，代入，指し示すデータの参照，整数値との加算と減算，およびポインタ同士の比較だけなのです（乗除算がないことに注意)。代入と参照についてはすでに説明しましたから，ここでは残りを説明します。

ポインタに対する四則演算としては整数値との加減算のみが許されています。ただし，この加減算は通常の場合と少し異なっています。ポインタはメモリ上に格納された何バイトかの大きさのデータの先頭アドレスを指し示しています。つまり，int型のデータを指し示すポインタなら，それが保持するひとつのアドレスで連続する4バイトのメモリを指し示していることになります。このとき，ポインタに1を加えるとはどういうことを意味するのでしょう。

ポインタの指し示すデータが4バイトの大きさを持つとき，先頭から1バイト先のところのデータを指し示しても，それは無意味なデータを指し示すだけです。+1という演算に何か意味を持たせるためには，ポインタが現在メモリ上で指し示しているデータの次のデータを指し示すようにするのが順当です(つまり4加える)。当然，C言語のポインタも+1という操作に関してこのような動きをするようになっています。逆にポインタの値を1減少させるときはメモリ上の1つ前のデータを指し示すようになっています。図3にこの概念図を示しておきます。これをまとめると，ポインタ変数に±1を加えるとき，

char型へのポインタなら	±1
short型へのポインタなら	±2
int型へのポインタなら	±4
float型へのポインタなら	±4
double型へのポインタなら	±8

だけポインタの保持しているアドレスが変化するということになります。これはポインタが指し示すデータ型の大きさ（バイト数）と等しくなっているのがわかりますね。これで何かを思い出しませんか。そう配列ですね。配列は添字が1増加することでひとつ次の配列要素，添字が1減少することでひとつ前の配列要素を表すのでした。ポインタに整数値の加減算は配列に対する添字のように働くのです。これはポインタを配列要素を指し示す変数として利用する可能性を表しています。具体的に説明しましょう。いま，

```
int a [5]= {1, 2, 3, 4, 5};
int *p = &a [2];
```

という宣言があると仮定します。int型データを指し示すポインタpは配列aの3番目の要素のアドレスに初期化されています。このとき，pを用いた式と配列のアドレスとの関係は，

```
p-2 →    &a [0]
p-1 →    &a [1]
p   →    &a [2]
p+1 →    &a [3]
p+2 →    &a [4]
```

という関係になります。つまり，pを連続的に変化させることはその指し示す先であるint型の配列要素を順にたどっていくことに等しいのです。当然，ポインタが指し示すデータについても，

```
*(p-2) →    a [0]
*(p-1) →    a [1]
*p     →    a [2]
*(p+1) →    a [3]
*(p+2) →    a [4]
```

という関係になっています。このようにポインタはメモリ上に連続して並んだ同じデータ型のデータに対し，ある位置から何番目という相対番号を指定してその要素を参照するのに役立ちます。

ここで，[ ] という演算子と＊という演算子の関係について考えましょう。[ ]はある基準アドレス(配列名)から指定した個数だけあと（前の場合もある）のデータを参照するための演算子です。たとえば，a [2] はaという配列名（アドレス）を基準にして3番目の要素（添字は0から始まるので)を参照します。この演算子はポインタに対しても同様に機能します。ポインタに対して [ ] を適用したときはポインタの保持するアドレスを基準にして指定した個数だけ前後の要素を参照するのです。つまり，ポインタpに関してp [3] は＊(p+3) とまったく同じ意味になります。逆にaが（特に1次元）配列名のときa [3] と＊(a+3) は同じ意味です。なんとなくポインタと配列名との関係がわかってきたような気がしませんか。

実はポインタと配列名とはただ1点の違いを除いて(プログラム上は)同じものです[8)]。ポインタは変数ですからアドレスを格納する領域，つまり実体があります。しか

し，配列名のほうはコンパイル時に決定されるメモリのアドレスを示す定数値で実体はありません（値はコンパイラだけが知っている)。したがって，ポインタは代入などにより値を変更することができます。しかし，配列名に対応するアドレス値を変更することは不可能です（変更すべき実体がない)。

ポインタで配列を指し示すことができることがわかりました。2つのポインタが同一の配列の要素を指し示しているとき，そのポインタ間では比較演算を定義することができます。つまり，

```
== != <= >= < >
```

という関係は正常に成立します。この大小関係は添字の大小関係と一致しています。ただし，お互いに無関係なデータ（異なる配列の要素や変数）を指し示している2つのポインタの比較には意味がありません（エラーにはならないが)。

また，ポインタは定数0と比較することができます。ポインタの値はメモリのアドレスですから，それが0であることはまずありません。X68000で0番地を参照すると間違いなくバスエラーです。それなのにポインタを0と比較する意味はエラーのチェックです。C言語の常識（？）として，ポインタを戻り値とするような関数ではエラーが発生したときは定数0を戻り値として返すようになっているからです。したがって，ポインタと0の比較は一致/不一致だけが重要であって，大小関係には意味がありません[9]。

さて，ポインタの特徴は，配列とほとんど同じことができる上に，その値を変更できるという点です。ポインタ変数には代入および加減の演算が許されますから，=，++，--，+=，-=といった演算子を使ってポインタ変数の指し示す位置をプログラマに都合のいいように変更できるのです。いま，上のaという配列の要素をすべて加えるプログラムを考えましょう。これは，自然な記述では添字による配列の参照を用いて，

```
for(s=0, i=0 ; i<5 ; i++)
    s += a [i] ;
```

となるでしょう。しかしこれはポインタ変数を用いて，

```
p = a;
q = a+5 ; /* 終わりのアドレス */
s = 0 ;
while(p<q) s += *p++ ;
```

と記述することもできます。この2つのプログラムの本質的な差は，配列要素の参照をa [i]で行うか*pで行うかです。これをCPUの実行という観点から考えてみると，a [i]は参照を行うたびに(a+i)というアドレス計算を行わなければなりません。しかし，pは次の要素を指し示すように常に更新されていますから，要素を参照するときにはいちいちアドレスを計算する必要はありません。このため，後者のポインタを使う例のほうが高速な実行ができるのです。このように，ポインタはプログラムの実行速度を速くするという利点があるのです。

最後にポインタへのキャスト（型変換）について説明しましょう。あるデータ型を指し示すポインタを別のデータ型を指し示すポインタとみなしてその指し示すデータを参照したいときがあります。たとえば，double型のデータを指し示すポインタをint型のデータを指し示すポインタと見なして内容を参照する必要があるかもしれません。このときは，double型のデータを指し示すポインタをint型のデータを指し示すポインタにキャストして内容を参照すればいいのです。キャストは目的のデータ型を(と)で囲って変数名の前につけることで実現します。変数の宣言に*をつけたのがポインタの宣言でした。ポインタへのキャストはデータ型のあとに*をつけます。たとえば，int型のデータを指し示すポインタへキャストは(int *)という演算子によって行います。したがって，xがdouble型のデータを指し示すポインタは，

```
(int *)x
```

と記述することでxの指し示すdouble型データをint型データとみなして参照することができます。

このようなキャストが有効なのは，char型の配列の要素をint型の配列と見なして0でクリアするときでしょう。たとえば，

```
char a [100] ;
int *p=(int*)a;
for(i=0 ; i<100/4 ; i++)
        *p++ = 0 ;
```

というプログラムでは，4バイト分の配列要素を同時に0にすることができるので，理論的には4倍速い速度で実行できるはずです。

---

8) &，*，[]などという演算子を配列名やポインタに適用するときはコンパイラがうまくやってくれるので気にならないが，配列とポインタに関する参照の差異は注意する必要がある。配列名は定数なのでそのまま使用できるが，ポインタは変数なのでその値（アドレス）を1回読み出す操作が必要になる。たとえば，*(p+3)という式のコンパイラでの処理がpが配列名かポインタかでどのように異なるか考えてみよう。

9) GCCではポインタと0の大小比較すると警告メッセージが出る。そういうプログラムを書くほうも書くほうだが，こんなものまでチェックしているとは頭が下がる。

◆基礎力を高めよう

設問1 ポインタへのポインタである変数aが，

```
char **a;
```

と宣言されているとき，次の1)～5)の値がいくつになるか答えてください。

1) sizeof(a)
2) sizeof(*a)
3) sizeof(a[2])
4) sizeof(*a[2])
5) sizeof(**a)

設問2 変数の宣言が，

```
char *a;
char b [10];
int ii;
char ci;
float fi;
```

となっているとき，次の1)～4)の式のうち文法的に許されるのはどれか答えてください。

1) ＊(a＋b)
2) ＊(a＋ii)
3) ＊(a＋ci)
4) ＊(a＋fi)

設問3 次のプログラムのうちで文法的に誤っているところを指摘してください。

```
char *str1="abcd";
char str2 [ ]="ABCD";
main( )
{
    str1="asdfg";
    str2="yxz";
    func(str1, 0);
}
func(s1, s2)
char *s1;
char s2 [ ];
{
    s1="1234";
    s2="qwerty";
}
```

(解答は128ページ)

ポインタを利用するプログラム

ポインタを利用するプログラムとして今回はクイックソートを取り上げます。クイックソートは通常は配列を扱うプログラムの例として取り上げられることが多いのですが，ポインタと配列はほとんど同じものだということで，今回はポインタを用いた版を書いてみました。

クイックソートとは，配列の要素をある添字を境として基準データより小さいものと大きいものが集まるように大まかな並び替えを行った後，それぞれの部分配列に関して同じ操作を再帰的に行うものです。これは，まじめに比較・交換を繰り返すソーティングアルゴリズムの中では，現時点でもっとも速いものといわれています。

クイックソートのプログラムをリスト1に示します。プログラムの解説はほとんど不要でしょう。通常は配列で行っているクイックソートのプログラムをポインタを使って書き直しているだけです。ただし，基準データは配列の中央の要素から持ってくるのが普通ですが，リスト1では配列の最初の要素を基準データとしています。配列の要素はまだ並び替えられていないのでどこを基準に選んでも同じはずですからね。

さて，リスト1だけではつまらないので，リスト2のプログラムを作ってみました。一見，リスト1と同じプログラムに見えますが，Quicksort関数の引数の宣言がchar型データを指し示すポインタへのポインタになっていま

リスト1

```
 1: /*
 2:      ポインタを利用したクイックソート（その１）
 3:
 4:      普通は配列を使ったプログラムの方が多いかも
 5: */
 6:
 7: int array[100]; /* ソートするデータ */
 8:
 9: InitArray() /* 乱数でソートするデータを作る */
10: {
11:      int i;
12:      for(i=0;i<100;i++)
13:           array[i]=rand();
14: }
15:
16: PrintArray()     /* データのプリント */
17: {
18:      int i;
19:      for(i=0;i<100;i++){
20:           printf("%8d ",array[i]);
21:           if((i%10)==9) printf("¥n");
22:      }
23:      printf("¥n");
24: }
25:
26: CheckResult()     /* ソート結果のチェック */
27: {
28:      int i;
29:      for(i=0;i<100-1;i++)
30:           if(array[i]>array[i+1])
31:                printf("結果がおかしい¥n");
32: }
33:
34: Quicksort(L0,R0)     /* クイックソート */
35: int *L0,*R0;
36: {
37:      int *L=L0,*R=R0;
38:
39:      do {
40:           while ( *L  < *L0 ) L++;
41:           while ( *L0 < *R  ) R--;
42:           if ( L<=R ) {
43:                int w;
44:                w = *L;
45:                *L++ = *R;
46:                *R-- = w;
47:           }
48:      } while ( L <= R );
49:      if ( L0 < R  ) Quicksort(L0,R);
50:      if ( L  < R0 ) Quicksort(L,R0);
51: }
52: /*******************************************************/
53: main()  /* メインプログラム */
54: {
55:      srand(time(0)); /* 乱数の初期化 */
56:      InitArray();    /* データを作る */
57:      PrintArray();   /* データの初期値をプリント */
58:      Quicksort(&array[0],&array[100-1]);
59:      PrintArray();   /* ソートした結果をプリント */
60:      CheckResult();  /* ソートした結果をチェック */
61: }
```

リスト1の実行結果

```
   31871     9078    23845     5964    28075     6658    25617     4504    22103    17422
   14973    22372    26243    20890     4201     6320    20015     7568    23056    31376
    7952    26512    29712    25744    12004    31238    14392    20986    29644    11822
    5536    25250     4532      854     7176    25674    10396     6526    27504    30450
   22660    26044    24477    17798    24503    30832    27633    14458     3469     4851
    4489    17103    11333    32619    31169     6621     4546    22991    31556     2401
   29030    19091     9768    24421    27530    21697    30663     8445    29155     6137
   11455     5557     4099    18205    26887    23617    15179    29861    27343     5383
   26692    24169    22198    27243    15560    19213    15482    12879     6092    18247
   27779     1951    10139    21367    18995    17103    21945     6272    23679      886

     854      886     1951     2401     3469     4099     4201     4489     4504     4532
    4546     4851     5383     5536     5557     5964     6092     6137     6272     6320
    6526     6621     6658     7176     7568     7952     8445     9078     9768    10139
   10396    11333    11455    11822    12004    12879    14392    14458    14973    15179
   15482    15560    17103    17103    17422    17798    18205    18247    18995    19091
   19213    20015    20890    20986    21367    21697    21945    22103    22198    22372
   22660    22991    23056    23617    23679    23845    24169    24421    24477    24503
   25250    25617    25674    25744    26044    26243    26512    26692    26887    27243
   27343    27504    27530    27633    27779    28075    29030    29155    29644    29712
   29861    30450    30663    30832    31169    31238    31376    31556    31871    32619
```

リスト2

```
 1: /*
 2:      ポインタを利用したクイックソート（その２）
 3:
 4: */
 5:
 6: char *array[100];    /* ソートするデータ（ポインタの配列） */
 7: char string[100][9];     /* 文字列を格納する配列 */
 8:
 9: InitArray() /* 乱数でソートするデータを作る */
10: {
11:      int i,j,len;
12:      for(i=0;i<100;i++){
13:           len=rand()%8+1; /* 文字列の長さ（１～８） */
14:           for(j=0;j<len;j++)
15:                string[i][j]='A'+(rand()%26);
16:           string[i][j]=0; /* １つの文字列 */
17:           array[i]=&string[i][0]; /* 文字列のアドレス */
18:      }
19: }
20:
21: PrintArray()     /* データのプリント */
22: {
23:      int i;
24:      for(i=0;i<100;i++){
25:           printf("%-8s ",array[i]);
26:           if((i%10)==9) printf("¥n");
27:      }
28:      printf("¥n");
29: }
30:
31: Compare(s1,s2)  /* 比較に使う関数 */
32: char *s1,*s2;    /* s1>s2 で正，s1==s2 で０，s1<s2 で負を返す */
33: {
34:      return( strcmp(s1,s2) );     /* 普通の場合 */
35: /*
36:      return( strlen(s1)-strlen(s2) ); なんてのも考えられる
37: */
38: }
39:
40: CheckResult()   /* ソート結果のチェック */
41: {
42:      int i;
43:      for(i=0;i<100-1;i++)
44:           if(Compare(array[i],array[i+1])>0)
45:                printf("結果がおかしい¥n");
46: }
47:
48: Quicksort(L0,R0)     /* クイックソート */
49: char **L0,**R0;
50: {
51:      char **L=L0,**R=R0;
52:
53:      do {
54:           while ( Compare(*L,*L0)<0 ) L++;
55:           while ( Compare(*L0,*R)<0 ) R--;
56:           if ( L<=R ) {    /* ポインタを交換するだけ */
57:                char *w;
58:                w = *L;
59:                *L++ = *R;
60:                *R-- = w;
61:           }
62:      } while ( L <= R );
63:      if ( L0 < R  ) Quicksort(L0,R);
64:      if ( L  < R0 ) Quicksort(L,R0);
65: }
66: /*******************************************************/
67: main() /* メインプログラム */
68: {
69:      srand(time(0)); /* 乱数の初期化 */
70:      InitArray();     /* データを作る */
71:      PrintArray();    /* データの初期値をプリント */
72:      Quicksort(&array[0],&array[100-1]);
73:      PrintArray();、 /* ソートした結果をプリント */
74:      CheckResult();  /* ソートした結果をチェック */
75: }
```

リスト2の実行結果

```
LEBCLEN  FGPEBAF  UUWSUEY  AKMCKKY  MSEKAKW  XWZYLGL  BVDL     FK       AJYBEDCX BNDXZHBB
PJ       TMVSDMJO MPRLFXVF EDYJ     ZAPGL    HOZ      BLAN     H        FWN      PCNAHUV
X        BUX      HLZNHNH  MNYDIF   SII      IGGSE    XQJAKUK  IGM      IMV      HQNCBCZ
AUA      N        XPRHD    ZPRH     VDDJQJOU OMF      PIZCFFD  FBPR     FLNX     WZWX
DNSDYX   N        KYA      A        E        SGHMP    BGJ      TQPXJNG  KTFJZBRR B
ZMZSMGG  IUIUCWE  MLRHJU   EFABWV   JMT      JQZCLMF  P        BHG      UJYDFD   VBNJ
PMXWX    FIV      NHZIRO   MH       HON      ZXPFXD   ZHOZGLPT KCGCKK   OEIOWYG  YNRXRD
PIVTBXG  YJ       HELYHGJ  YIGIVKH  ONOJMXBU YZ       R        DQQ      SYY      OGR
PPL      UZCZHVHB RWXO     IVMAZCBO WZPL     KZ       IHQKCGQ  IMYXQFS  SHCBSNMV S
PFP      BNNYIY   YOU      FJJBVJ   XZIF     HHCLMCQ  S        A        SPWRUJKF SQU

A        A        AJYBEDCX AKMCKKY  AUA      B        BGJ      BHG      BLAN     BNDXZHBB
BNNYIY   BUX      BVDL     DNSDYX   DQQ      E        EDYJ     EFABWV   FBPR     FGPEBAF
FIV      FJJBVJ   FK       FLNX     FWN      H        HELYHGJ  HHCLMCQ  HLZNHNH  HON
HOZ      HQNCBCZ  IGGSE    IGM      IHQKCGQ  IMV      IMYXQFS  IUIUCWE  IVMAZCBO JMT
JQZCLMF  KCGCKK   KTFJZBRR KYA      KZ       LEBCLEN  MH       MLRHJU   MNYDIF   MPRLFXVF
MSEKAKW  N        N        NHZIRO   OEIOWYG  OGR      OMF      ONOJMXBU P        PCNAHUV
PFP      PIVTBXG  PIZCFFD  PJ       PMXWX    PPL      R        RWXO     S        S
SGHMP    SHCBSNMV SII      SPWRUJKF SQU      SYY      TMVSDMJO TQPXJNG  UJYDFD   UUWSUEY
UZCZHVHB VBNJ     VDDJQJOU WZPL     WZWX     X        XPRHD    XQJAKUK  XWZYLGL  XZIF
YIGIVKH  YJ       YNRXRD   YOU      YZ       ZAPGL    ZHOZGLPT ZMZSMGG  ZPRH     ZXPFXD
```

す。これは文字列を指し示すポインタを要素とする配列を受け取って，そのポインタの指し示す文字列に関する並び替えを行います。このプログラムの特徴は文字列の交換をポインタを交換することで行っています。これにより，strcpy関数などで文字列全体を転送して交換する必要がなくなり高速な処理が実現できます。なお，文字列の大小比較はCompareという関数で行っていますからこれを適当に入れ替えることでいろいろな並び替えをすることができます。

リスト1，2のプログラムはアドレス参照あり，ポインタの参照あり，ポインタの比較ありで，ポインタを復習するにはちょうどいいかもしれませんね。なお，リスト1，2の実行結果は乱数で作成した配列を並び替えているので同じ結果になるとは限りませんのであしからず。

おわりに

いまさら説明をするまでもなく，C言語の学習においてポインタは最大難関です。そしてポインタはC言語の連載においても最大のテーマです。ポインタをどう解説するかで連載の価値が決まってしまうといってもよいでしょう。

この連載を始めた頃からポインタに関する説明をどうしようかと考えていましたが，悩みに悩んだ挙げ句，今回のようになってしまいました。あっと驚くような説明をすることはできませんでしたが，基本的なところは押さえることができたと思います。やはり，配列とポインタの類似性，違いをしっかりと理解することが必要だと思います。気長に学習していきましょう。

＜参考文献＞

1）　柴田望洋，『秘伝C言語問答ポインタ編』，ソフトバンク，1991年．

2）　R.J.Traister（新井利幸訳），『Mastering C Pointers--C言語のプログラミング・パワーを100%引き出す』，HBJ出版局，1991年．

3）　祝一平，『K＆Rも知らないC言語のひ・み・つ』，Oh!X 1989年7月号，pp.30-31．

◆基礎力を高めようの解答

設問1

1）4　2）4　3）4　4）1　5）1

解説

X68000のCコンパイラではポインタの大きさは4バイトである。ポインタaは＊を2個つけて初めてchar型のデータになる。[　]も効果は＊と同じ。

設問2

1）×　2）○　3）○　4）×

解説

ポインタと加減算できるのは整数のみ。これは配列の添字に関しても同じ。

設問3

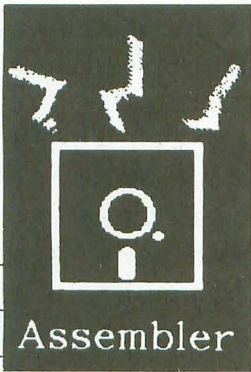
str2＝"yxz"; の部分が誤り。

解説

配列名はアドレスを示す定数なので値を変更することはできない。ただし，関数の引数の宣言ではポインタも配列も同じものになる。したがって，引数で宣言した配列名には値を設定することができる。なお，funcという関数は引数に文字列のアドレスをコピーするが，抜けるときにその値を捨てるので，結果として何もしない。

グラフィックパターンの回転

Murata Toshiyuki 村田 敏幸

前回，拡大・縮小をかたづけたいきおいで，今回はグラフィックパターンの回転に取り組みます。村田氏いわく“真面目に回転ルーチンを作るのは初めて”ということですが，結果として満足のいく仕上がりになったようです。

今回のテーマはグラフィックパターンの回転だ。汎用の回転つきプットルーチンを作成する。今月のプログラムは僕が作る2本目の回転ルーチンとなる。1本目はその昔，256×256ドット65536色の画像をぐるぐる回すデモプログラムを初めて見た日に，悔しかったもんだから朝までかかって作った。いまでこそパターンの回転ぐらい珍しくもないが，当時は結構インパクトがあったのだ。

もっとも，あのとき作ったのは本当に256×256ドットの画像を回転させる機能しかなく，しかも，回転の計算は先にCだったかX-BASICだったかにやらせマシン語部はその計算結果を利用するだけという作りだったから，真面目に回転ルーチンを作るのは今回が初めてだったりする。うまく動いたらほめてほしいもんだ。

X-BASICで描く

理屈の面から攻めよう。図1を見てもらいたい。この図から，点(x, y)を原点回りに角度θだけ回転する座標変換式が得られる。移動先の点(x′, y′)はつぎの式で与えられることがわかる。

$$x' = x \times \cos\theta - y \times \sin\theta$$

$$y' = x \times \sin\theta + y \times \cos\theta$$

試しにこの式を使ってX-BASICで回転ルーチンを書いてみるとリスト1のようになった。リスト1は画面左上に描いた64ドット四方の適当な絵を30度回転して画面中央に描き直す。回転に先立って座標から32を引いている点と，回転角度が30度ではなく−30度になっている点をチェックしておいてほしい。前者は回転の中心を画像の中央にとるためで，後者はグラフィック画面と数学の座標系とではy座標軸の向きが逆になっていることを反映している。

リスト1は大筋ではうまく動く。だが，回転後の画像の質はあまりよくない。いびつに歪むし，黒い点がポツポツと残る。

歪みの原因はピクセルの縦横比にある。表示画面が512×512や256×256のとき，ピクセルの形は縦横比約2：3の横長であり，見た目に自然な回転結果を得るにはこの点を考慮する必要がある。具体的には，x座標を1.5倍してから回転の計算をし，回転後，得られたx座標を2/3に再修正する。

図1

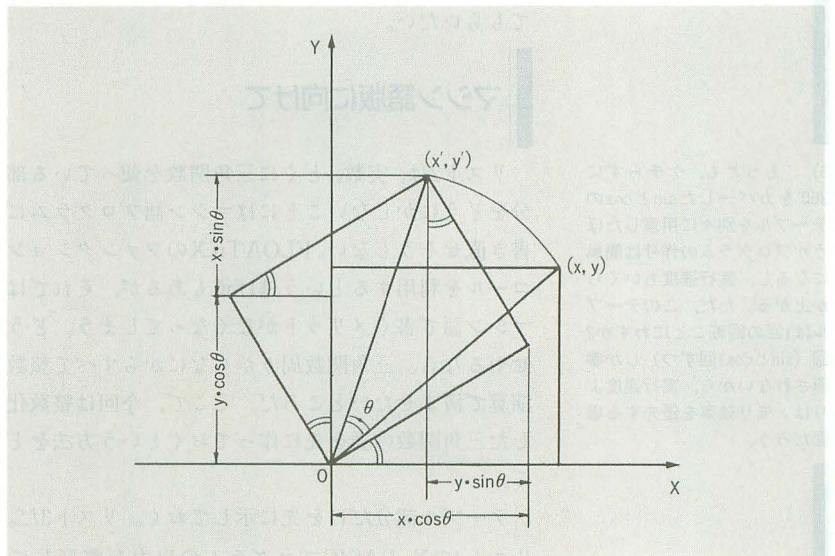
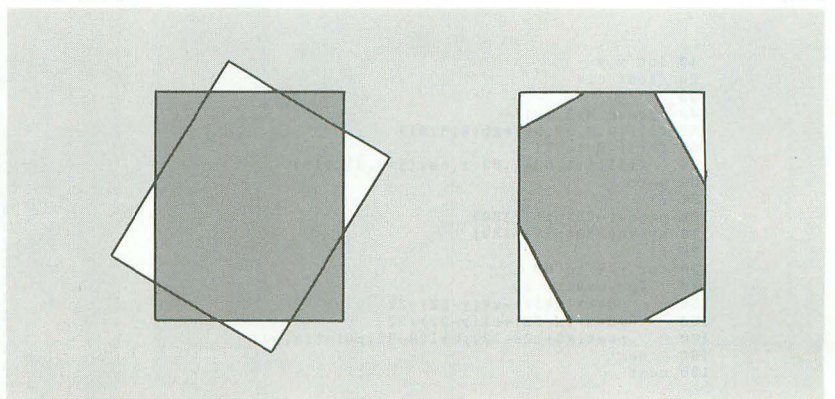



図2

黒い点が残るのは，グラフィック画面上では斜めの線が正確には表現できないためだ。斜線は階段状になり，その段がきれいに噛み合わない部分に隙間が生まれる。この穴は先月の拡大・縮小のときと同じ逆転の発想で埋めることができる。(x, y)を回転して(x′, y′)を求める代わりに，(x′, y′)を逆回転することで対応する変換元の位置(x, y)を求めるという考え方だ。元画像を傾けて貼り付けるのではなく，回転後水平になるよう画像を切り出してくると思えばよい（図2)。この場合，回転によってはみ出す角の部分は描画されず，その代わり元画像外周の余白も一緒に回転して描かれることになる。プットルーチンにおいて余白はグラフィックパターンを格納したメモリの外側を意味するから，実際のプログラムでは余白ができないよう，グラフィックパターンをひと回り大きく用意することになるだろう。

以上2点に注意して作り直したのがリスト2だ。隙間は埋まり，歪みもだいぶなくなっている。ピクセルの縦横比が正確に2：3ではないため補正が不十分で（というかやりすぎ）いびつさが消えてはいないが，なんとか許容範囲だと思う。気になるなら，より自然な結果が得られるよう補正値を微調整してみてもらいたい。

マシン語版に向けて

リスト2は，実数，とくに三角関数を使っている部分をどうにかしないことにはマシン語プログラムに書き直せそうもない。FLOATn.Xのファンクションコールを利用するという逃げ道もあるが，それではマシン語で書くメリットがなくなってしまう。どうせやるなら，三角関数周りからなにからすべて整数演算で済ませたいところだ。そこで，今回は整数化した三角関数の表を先に作っておくという方法をとる。

テーブル部分だけを先に示しておく。リスト3だ。リスト4のX-BASICプログラムの出力を整形して作成した。0°～90°まで1°きざみで$\sin\theta$を4000_H倍した値を並べてある[1]。$\sin\theta$倍するときにはθに対応する値をこのテーブルから取り出して，掛けたのち，4000_Hで割る。あくまで近似計算だが，今回のプログラムで使う分には十分な精度が出る。

ここで，4000_Hという数字を選んだのにはちゃんと理由がある。0～1(sin, cosの値域)の値と掛け合わせても16ビット符号付き数に収まるなるべく大きな2のn乗の値，ということで決めた。16ビット符号付き数に収めるのはあとでmulsを有効に使うため，値を大きくするのは精度を上げるため，そして，2のn乗にするのは除算がシフトで済むようにするためだ[2]。

さて，リスト3にはsinだけ，それも90°までの狭い範囲の値しか用意していない。cosを求める場合や角度が0°～90°の範囲外のときには，三角関数の性質を利用して対応する。cosはsinの位相を90°ずらしたものだから，

$$\cos\theta=\sin(90°-\theta)$$

により得られる。また，角度θは，

$$\sin(-\theta)=-\sin\theta$$

$$\sin(180°-\theta)=\sin\theta$$

の関係から0°～90°に収めることができる。たとえば，

$$\sin(-150°)=-\sin(150°)=-\sin(180°-30°)=-\sin30°$$

となる[3]。

sin, cosが求められるようになったことで，マシン語で回転ルーチンを書く目途が立った。安心して，効率の向上を目指そう。リスト2のX-BASICプログラムでは1ピクセルごとに回転の計算をしているが，この計算回数をなるべく減らす方向で考えてみる。

斜めに点を拾って水平に並べることで回転を実現していることに注目する。水平1ラインを描く過程で拾う点は（グラフィック画面で表現できる精度の）直線上に位置しているわけだ。ということは，始めの点と最後の点が決まれば，Bresenhamの線分発生

1） sinの値を小数部14ビットの固定小数点で表現しているといういい方もできる。

2） 4000_Hではなく8000_H倍を採用してさらに精度を稼ぐという選択肢もある。sin90°＝1だから8000_H倍するとオーバーフローして－32768に化けてしまうが，値が1だというのがわかっているのだから，特別扱いすればよい。

3） もっとも，ケチらずに360°をカバーしたsinとcosのテーブルを別々に用意したほうがプログラムの作りは簡単になるし，実行速度もいくらか上がる。ただ，このテーブルは1回の回転ごとにわずか2回（sinとcos1回ずつ）しか参照されないから，実行速度よりはメモリ効率を優先する場面だろう。

リスト1

```
10 int x,y
20 float c,s
30 /*
40 screen 0,3,1,1
50 fill(0,0,99,99,rgb(8,8,8))
60 for i=0 to 31
70   fill(i,i,63-i,63-i,hsv(i*6,31,31))
80 next
90 /*
100 c=cos(-30*pi()/180)
110 s=sin(-30*pi()/180)
120 /*
130 for y=0 to 63
140   for x=0 to 63
150     x0=c*(x-32)-s*(y-32)+32
160     y0=s*(x-32)+c*(y-32)+32
170     pset(x0+128-32,y0+128-32,point(x,y))
180   next
190 next
```

リスト2

```
10 int x,y,rx=3,ry=2
20 float c,s
30 /*
40 screen 0,3,1,1
50 fill(0,0,99,99,rgb(8,8,8))
60 for i=0 to 31
70   fill(i,i,63-i,63-i,hsv(i*6,31,31))
80 next
90 /*
100 c=cos(30*pi()/180)
110 s=sin(30*pi()/180)
120 /*
130 for y=0 to 63
140   for x=0 to 63
150     x0=ry*(rx*c*(x-32)/ry-s*(y-32))/rx+32
160     y0=rx*s*(x-32)/ry+c*(y-32)+32
170     pset(x+128-32,y+128-32,point(x0,y0))
180   next
190 next
```

アルゴリズムの応用で，そのあいだのどの点を拾ったらよいかがわかる。しかも，各水平ラインに対応するグラフィックパターン上の参照開始位置もまた直線上に並ぶから，やはり線分発生アルゴリズムにより順次求めることができる。結局，回転の計算は矩形領域の角3点に対してだけ行えば十分だということがわかる。この様子を図3に示した。

ここで，図3の点Bからbへの回転による移動距離とDからdへの移動距離は符号が違うだけで“ほぼ”等しい[4]ことを利用すると，一方の回転計算を手抜きできる点も指摘しておこう。

さて，全ピクセルについて回転の計算をしていたのが，2.5点分の計算で済むところまで簡略化できた。が，この考え方を実用に供するにはもうひと工夫いる。グラフィック画面上では数学的な意味での線分の長さとピクセル数が一致せず，斜めの線分は同じ長さの水平/垂直線分よりもピクセル数が少ないのだ。グラフィックパターンを斜めに拾っただけでは，描画先の水平ラインを埋めるだけのピクセル数がない。したがって，適当に引き伸ばす必要がある。斜めに拾った点の集まりを細長いパターンに見立て，先月のプログラムと同じ方法で拡大するのだ。

最終的に，回転プログラムは，斜めに点を拾うのにBresenhamのアルゴリズムを利用し，それを拡大するのにもBresenhamのアルゴリズムを使い，これを縦方向と横方向に独立して行うという構成で実現される。

回転つきプットルーチンの作成

では，回転つきプットルーチンをリスト5に示そう。回転処理過程でのパターン引き伸ばし部分に少々融通をきかせた結果，回転だけではなく拡大・縮小も同時に行うことができる贅沢なサブルーチンになった。どのみち拡大の処理は必要なのだから，このように拡張してもプログラムの複雑さは変わらない。ただ，拡大・縮小はあくまでおまけにすぎず，演算過程で比較的簡単にオーバーフローを起こすので，正しく拡大・縮小できる範囲は狭い。

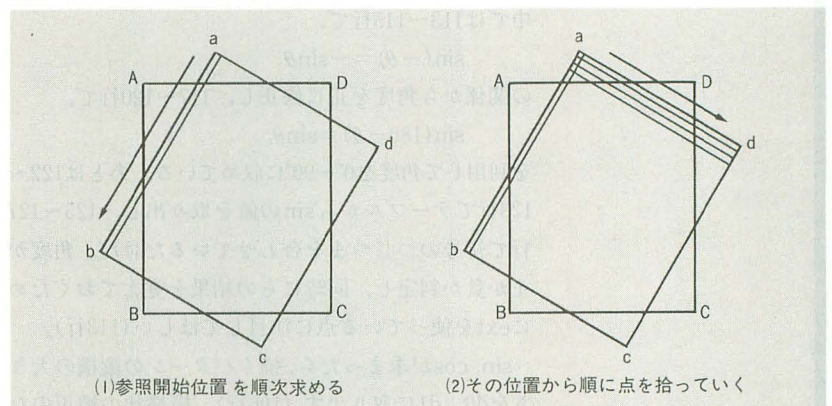
リスト5のサブルーチンgrputには13〜21行のような構造で用意した引数列へのポインタをスタックに積んで渡す。パターンとその大きさの指定の仕方にはちょっと癖がある。パターンは余白を計算に入れて，実際に描く範囲の縦横2倍の大きさで用意するのだが，引数中のXL，YLは2倍しない値で表す。また，PATはパターンの先頭アドレスではなく，“実際に描く範囲の左上隅”を指す仕様になっている。この関係はリスト中の23〜31行に示した。あと，少々手抜きがあり，パターンの縦横の大きさはなるべく近い値でなければならない。このプログラムはパターンが細長いと回転の過程でパターン外のメモリを平気でアクセスしてしまうのだった。

末尾の引数TEMPは作業用のメモリへのポインタだ。grputは処理速度向上のため，前回の拡大・縮小ルーチン同様，描画に先立ってテーブルを作成する。作業用に使うメモリ量は描画範囲によっても変わるが，あとで施す高速化を計算に入れても，256×256の範囲に描画する場合で，4Kバイトもあればよい。

簡単な動作試験用のプログラムをリスト6に用意しておく。適当な65536色モードの画像を画面に読み込んでから実行すると，中央の256ピクセル四方を取り込み，1°単位で回転させる(仕様上，実際に描かれ

4)　理論上は“等しい”のだが，回転するパターンの大きさが偶数ピクセルの場合，回転の中心となるピクセルが正確には決まらないため誤差が出る。

図3

(1)参照開始位置を順次求める　　(2)その位置から順に点を拾っていく

リスト3

```
 1: *sin 0° ～ 90° の表 (16384倍)
 2:
 3:         .xdef   sintable
 4:
 5:         .text
 6: *
 7: sintable:
 8:         .dc.w   00000,00286,00572,00857,01143   *0
 9:         .dc.w   01428,01713,01997,02280,02563
10:         .dc.w   02845,03126,03406,03686,03964   *10
11:         .dc.w   04240,04516,04790,05063,05334
12:         .dc.w   05604,05872,06138,06402,06664   *20
13:         .dc.w   06924,07182,07438,07692,07943
14:         .dc.w   08192,08438,08682,08923,09162   *30
15:         .dc.w   09397,09630,09860,10087,10311
16:         .dc.w   10531,10749,10963,11174,11381   *40
17:         .dc.w   11585,11786,11982,12176,12365
18:         .dc.w   12551,12733,12911,13085,13255   *50
19:         .dc.w   13421,13583,13741,13894,14044
20:         .dc.w   14189,14330,14466,14598,14726   *60
21:         .dc.w   14849,14968,15082,15191,15296
22:         .dc.w   15396,15491,15582,15668,15749   *70
23:         .dc.w   15826,15897,15964,16026,16083
24:         .dc.w   16135,16182,16225,16262,16294   *80
25:         .dc.w   16322,16344,16362,16374,16382
26:         .dc.w   16384                           *90
27:
28:         .end
```

リスト4

```
10 int fp, i, s
20 /*
30 fp = fopen("sintable.$$$","c")
40 for i = 0 to 90
50      s = int(sin(pi()*i/180)*16384+0.5#)
60      fwrites(itoa(s)+chr$(13)+chr$(10), fp)
70 next
80 fclose(fp)
```

るのは128ピクセル四方)。リスト6中，59行の表示範囲を適当に変更して，拡大・縮小の動作も確認しておいてもらいたい。

サブルーチンgrputはいくつかの下請けサブルーチンからなる。以下，順に見ていこう。

●サブルーチンrotate (171～246行)

rotateは座標回転の計算をする。172行で引数列から回転角度をd4に取り出し，174～184行でこの角度が－180°～＋179°に収まるように360を加減算する。186～190行では，あとでcosを求めるときのために90－d4をd5に計算している。角度の正規化が済んだら192行でsinのテーブルへのポインタをa0に入れ，193～194行でマクロSINを使ってsin(d4)，cos(d4)＝sin(d5)をテーブル参照により得る。

マクロSINは110～129行で定義してある。マクロ中では113～115行で，

$\sin(-\theta)=-\sin\theta$

の関係から角度を正に修正し，117～120行で，

$\sin(180-\theta)=\sin\theta$

を利用して角度を0°～90°に収めている。あとは122～123行でテーブルからsinの値を取り出し，125～127行で符号のつじつまを合わせているだけだ。角度が正か負か判定し，同時にその結果を覚えておくためにextを使っている点に注目してほしい(113行)。

sin，cosが求まったら，描くパターンの縦横の大きさをd0，d1に取り出す(196行)。縦横比の補正のため横方向の長さをすかさず1.5倍している(198～202行)。すでに触れたように，この2：3という比率は近似値だから，こだわりたい人は微調整を。

207～211行で回転の中心を求め，213～2[illegible]行でパターンの左上隅の点，221～223行で右上隅の点，225～228行で左下隅の点を回転する。左下の点については，右上の点の回転結果を使った簡略計算でごまかし，残り2点の回転計算は134～166行のマクロROTで実現している。ROTは汎用性のまったくないその場しのぎのマクロだが，とりあえず，リストの行数を減らすのには貢献してくれた。

ROT中では，140～143行で，

$x\times\cos\theta$

$y\times\cos\theta$

$x\times\sin\theta$

$y\times\sin\theta$

をレジスタに求め，この値を使って145～149行で回転後のx座標，151～155行でy座標を計算する。sin，cosの値はテーブル上では4000_H倍してあるため，得られた座標は4000_Hで割る(14ビット右シフトする)必要があることを思い出そう。リストではこのビットシフトを，2ビットの左シフトとswapで実現した。ここで，座標が負の場合のみシフトに先立って3FFF_Hを足しているのは，小数点以下の切り捨てが0に近づく方向に行われるようにするための処置だ。

ROTの最後，160～164行はピクセルの縦横比による再補正だ。1.5倍して計算していたx座標を2/3にして元に戻している。四捨五入したいがために，4倍してから3を足して6で割るという，少々冗長な手順を踏んでいる。

237行に達した時点で，パターンの右下を除く3つの角の回転結果が，

左上　(d0，d1)
右上　(d2，d3)
左下　(d4，d5)

に求まっている。サブルーチンrotateは最後に237～245行で，回転後の左上の点に対応するパターン上の位置をa1に計算し，d0～d5，a1をサブルーチンからの戻り値として返す。

●サブルーチンline_comp (252～305行)

line_compは線分(d0，d1)－(d2，d3)に沿ってグラフィックパターン上の点を拾うときのポインタの相対的な移動量をBresenhamのアルゴリズムによって求め，a5以下のメモリに並べてテーブルを作成する。ついでに，その線分上に位置するピクセル数をd0に返す。描画時にパターン上の点を斜めに拾う，その拾い方を決めるわけだ。

テーブルにはBresenhamのアルゴリズムを実行したときの，ポインタの変化の仕方(たとえば，始点から，右，右，右下，右，右，右下，……)をパターン上のアドレスの差の形で記録する。一度このテーブルができてしまえば，始点を指すポインタにテーブルの値を順次加えていくだけで，同じ傾き/長さの線分をなぞることができる。

line_compの中身は以前作ったラインルーチンの一番最初の版とそっくり同じ構造をしているから，解説は不要だろう。

●サブルーチンclip (325～424行)

clipはクリッピングを担当する。一度の呼び出しでx方向かy方向かのどちらか片方のクリッピングのみを行う。d1，d2で渡されるパターン左右端のx座標(あるいは上下端のy座標)をクリッピングウィンドウの大きさでクリップし，そのクリッピング量に応じてパターン側も同じ比率で切り捨てる。前回の拡大・縮小ルーチン同様，ここでは線分のクリッピングアルゴリズムを応用している。

パターン側のクリッピングは結構面倒臭い。パターンを斜めに切り捨てなければならないのだ。右端(あるいは下端)でのクリッピング時には何ピクセルはみ出したかがわかれば十分だが，左端(あるいは上端)でクリッピングする場合は，a1がポイントしているパターン上の参照開始位置も再計算する必

要がある。ここでは何ピクセル切り捨てるか求めたあと，line_compが作成したテーブルをそのピクセル数だけ頭から参照して移動量を合計し，クリッピング後の位置までポインタa1を移動している(370〜371行)。

●expand_comp（431〜453行）

expand_compはパターンを斜めに拾ってできるピクセル列を描画先領域の大きさに合わせて拡大・縮小するためのテーブルを作成する。考え方は前回の拡大・縮小ルーチンと同様だ。やはり，ポインタの相対的な移動量のテーブルを作る。

すでにline_compがパターンを斜めに拾うときのポインタの移動量をテーブルにしているので，expand_compの仕事はそのテーブルを適当に引き伸ばしたり圧縮したりするだけだ。拡大する場合は，同じピクセルが複数回参照されるようにすればよいわけだから，テーブルの適当なところに0を入れ引き伸ばす。縮小するときには，ポインタが斜め線上を飛び飛びに移動するよう，適当に隣り合った要素を足してまとめてテーブルを圧縮する。もちろん，引き伸ばしたり，圧縮したりするテーブル上の位置はBresenhamのアルゴリズムを使ってバランスよく決める。

●サブルーチンgrput（36〜105行）

描画ルーチン本体であるgrputは，サブルーチンrotate，clip，line_comp，expand_compを順次呼び出し，最終的にexpand_compが作成したテーブルだけを使って描画を行う。clip，line_comp，expand_compはx方向とy方向それぞれについて2回呼び出し，テーブルも2本作る。実際に描画を行う90〜100行のループでは，y方向のテーブルを使って始点を決め，x方向のテーブルを使ってそこから斜めに点を拾い描画する。

ところで，y方向のテーブルはただ一度しか使われないから，本来はテーブルを作らずに描画と並行してBresenhamのアルゴリズムを適用すれば済む。今回のプログラムではプログラムの簡潔さを優先し，y方向についてもテーブルを作成する形にした。

回転ルーチンの高速化

最後にリスト5をもう一段高速化してみる。水平1ラインの描画を行っている94〜96行のループに注目する。ループを展開すれば，

```
move.w  (a4), (a0)+
adda.w  (a5)+, a4
move.w  (a4), (a0)+
adda.w  (a5)+, a4
   ⋮
```

のようになるのはいうまでもないだろう。ここで，もし，回転角度や拡大・縮小率が固定ならば，この命令列は，

```
move.w  (a4), (a0)+
lea.l   n1(a4), a4
move.w  (a4), (a0)+
lea.l   n2(a4), a4
   ⋮
```

という，より高速なルーチンに置き換えることができる。n1，n2は本来はテーブルに並べてあったポインタの移動量だ。

ループをこのような形に展開できれば，1ライン分の描画はこれまでの1.5倍以上高速になると予想される。しかし，さすがに任意の回転角度，拡大・縮小率に応じた専用ルーチンを用意するのは不可能だ。そこで，grputの呼び出しごとに最適なルーチンを“作る”ことにする。x方向についてはポインタの移動量のテーブルを作る代わりに，描画ルーチン自体を生成するのだ。

リスト7が1ライン分の描画ルーチンを作成するサブルーチンだ。リスト5の最後に追加し，さらにリスト5の57行を，

```
bsr     expand_comp2
```

に，90〜100行を，

```
yloop: movea.l  a1, a4
       jsr      (a2)
       adda.w   (a3)+, a1
       adda.w   d0, a0
       dbra     d5, yloop
```

に変更して組み込む。

リスト7のサブルーチンexpand_comp2では，expand_comp同様の方法でポインタの移動量を求めては，

```
move.w  (a4), (a0)+
lea.l   XXX(a4), a4
```

のXXXの部分にその値を埋め込んだ3ワードの命令列を作業用メモリに書き込んでいく。最後に，

```
rts
```

の命令コードを書き込んで，1ライン分描画するサブルーチンの出来上がりだ。

なお，ページの都合で割愛したのだが，前回の拡大・縮小ルーチンも同様の方法で高速化することができる。気が向いたら試してみてもらいたい。

では，来月以降の予告をして終わろう。そろそろ仮称『グラフィックプリミティブ編』にケリをつける方向にもっていきつつある。予定している残りテーマは矩形領域の自由変形，円の描画，ペイント（順不同）の3回分だ。その先は白紙だから，プレッシャーをかけるならいまのうちかもしれない。

リスト5

```
  1: *          パターンを回転/拡大/縮小してプットする
  2:
  3:            .include        gconst.h
  4:            .include        gmacro.h
  5: *
  6:            .xdef   grput
  7:            .xref   gramadr
  8:            .xref   ucliprect
  9:            .xref   sintable
 10: *
 11:            .offset 0           *grputの引数構造
 12: *
 13: X0:        .ds.w   1           *描画先座標
 14: Y0:        .ds.w   1           *
 15: X1:        .ds.w   1           *
 16: Y1:        .ds.w   1           *
 17: DEG:       .ds.w   1           *回転角度(°)
 18: PAT:       .ds.l   1           *パターンアドレス(※)
 19: XL:        .ds.w   1           *パターンの横の長さ-1(※)
 20: YL:        .ds.w   1           *パターンの縦の長さ-1(※)
 21: TEMP:      .ds.l   1           *作業用メモリへのポインタ
 22: *
 23: *(※)           ┌  XL   ┐
 24: *          ┌─┬───┬─┐
 25: *          │  │      │  │
 26: *          ├─┼───┼─┤┐
 27: *          │  │＼    │  │
 28: *          │  │ pat  │  │YL
 29: *          ├─┼───┼─┤┘
 30: *          │  │      │  │
 31: *          └─┴───┴─┘
 32: *
 33:            .text
 34:            .even
 35: *
 36: grput:
 37: ARGPTR = 4+8*4+7*4
 38:            movem.l d0-d7/a0-a6,-(sp)
 39:
 40:            move.l  ARGPTR(sp),a6   *a6=引数列
 41:
 42:            bsr     rotate          *回転の計算をする
 43:
 44:            movea.l TEMP(a6),a5
 45:            movea.l a5,a4
 46:            movem.w d0-d1/d4-d5,-(sp)
 47:            bsr     line_comp           *水平1ライン分の
 48:                                        *  点の拾い方を決める
 49:
 50:            lea.l   ucliprect,a0    *左右端の
 51:            move.w  X0(a6),d1       *  クリッピングをする
 52:            move.w  X1(a6),d2       *
 53:            bsr     clip            *
 54:            bmi     done2           *
 55:
 56:            movea.l a5,a2
 57:            bsr     expand_comp     *水平1ライン分の
 58:                                    *  拡大/縮小を考慮した
 59:                                    *  点の拾い方を決める
 60:
 61:            movea.l a5,a4
 62:            movem.w (sp)+,d0-d3
 63:            bsr     line_comp       *垂直1ライン分の
 64:                                    *  点の拾い方を決める
 65:
 66:            lea.l   ucliprect+2,a0  *上下端の
 67:            move.w  Y0(a6),d1       *  クリッピングをする
 68:            move.w  Y1(a6),d2       *
 69:            bsr     clip            *
 70:            bmi     done            *
 71:
 72:            movea.l a5,a3
 73:            bsr     expand_comp     *垂直1ライン分の
 74:                                    *  拡大/縮小を考慮した
 75:                                    *  点の拾い方を決める
 76:
 77:            move.w  (a2)+,d0        *(d0,d1)=
 78:            move.w  (a3)+,d1        *  描画開始位置
 79:            jsr     gramadr         *a0=そのG-RAM上アドレス
 80:
 81:            move.w  (a2)+,d7        *d7=描画範囲横方向の長さ-1
 82:            move.w  (a3)+,d5        *d5=描画範囲縦方向の長さ-1
 83:
 84:            move.w  d7,d0           *d0=
 85:            addq.w  #1,d0           *  ライン右端と
 86:            add.w   d0,d0           *  次のライン左端との
 87:            neg.w   d0              *  G-RAM上のアドレスの差
 88:            addi.w  #GNBYTE,d0      *
 89:
 90: yloop:     movea.l a1,a4           *a4=参照するパターン上位置
 91:            movea.l a2,a5           *a5=1ライン分の参照位置の
 92:                                    *  移動量のテーブル
 93:            move.w  d7,d6
 94: xloop:     move.w  (a4),(a0)+      *1点描画する
 95:            adda.w  (a5)+,a4        *参照位置を移動する
 96:            dbra    d6,xloop        *横幅分繰り返す
 97:
 98:            adda.w  (a3)+,a1        *参照位置を縦に移動する
 99:            adda.w  d0,a0           *a0=次のライン左端
100:            dbra    d5,yloop        *高さ分繰り返す
101:
102: done:      movem.l (sp)+,d0-d7/a0-a6
103:            rts
104: done2:     addq.l  #8,sp
105:            bra     done
106:
107: *
108: *          sin(DEG)をテーブルから引いてくるマクロ
109: *
110: SIN        macro   TABLE,DEG
111:            local   skip1,skip2,skip3
112:
113:            ext.l   DEG             *sin(-θ)=-sin(θ)
114:            bpl     skip1           *
115:            neg.w   DEG             *
116:
117: skip1:     subi.w  #90,DEG         *sin(180°-θ)=sin(θ)
118:            bcs     skip2           *
119:            neg.w   DEG             *
120: skip2:     addi.w  #90,DEG         *0≦deg≦90°
121:
122:            add.w   DEG,DEG
123:            move.w  0(TABLE,DEG),DEG
124:
125:            tst.l   DEG             *sin(-θ)=-sin(θ)
126:            bpl     skip3           *
127:            neg.w   DEG             *
128: skip3:
129:            endm
130:
131: *
132: *          点(X,Y)を回転するマクロ
133: *
134: ROT        macro   X,Y
135:            local   skip1,skip2
136:
137:            move.w  X,d6
138:            move.w  Y,d7
139:
140:            muls.w  d5,X            *x・cos(θ)
141:            muls.w  d5,Y            *y・cos(θ)
142:            muls.w  d4,d6           *x・sin(θ)
143:            muls.w  d4,d7           *y・sin(θ)
144:
145:            sub.l   d7,X    *x・cos(θ)-y・sin(θ)
146:            bpl     skip1
147:            addi.l  #$3fff,X
148: skip1:     asl.l   #2,X
149:            swap.w  X
150:
151:            add.l   d6,Y    *x・sin(θ)+y・cos(θ)
152:            bpl     skip2
153:            addi.l  #$3fff,Y
154: skip2:     asl.l   #2,Y
155:            swap.w  Y
156:
157:            add.w   a2,X
158:            add.w   a3,Y
159:
160:            add.w   X,X             *x座標を2/3にする
161:            add.w   X,X             *
162:            addq.w  #3,X            *
163:            ext.l   X               *
164:            divs.w  #6,X            *
165:
166:            endm
167:
168: *
169: *          パターンの角3点を回転する
170: *
171: rotate:
172:            move.w  DEG(a6),d4      *d4=角度
173:
174:            move.w  #360,d6         *角度を
175:            move.w  #180,d7         *  -180°～179°に
176:            add.w   d7,d4           *  正規化する
177:            bpl     norm2           *
178: norm1:     add.w   d6,d4           *
179:            bmi     norm1           *
180: norm2:     cmp.w   d6,d4           *
181:            bcs     norm3           *
182:            sub.w   d6,d4           *
183:            bra     norm2           *
184: norm3:     sub.w   d7,d4           *
185:
186:            moveq.l #90,d5          *cosθ=sin(90°-θ)
187:            sub.w   d4,d5           *
188:            cmp.w   d7,d5           *
189:            ble     rot             *
190:            sub.w   d6,d5           *
191:
192: rot:       lea.l   sintable,a0
193:            SIN     a0,d4           *d4=sin(θ)
194:            SIN     a0,d5           *d5=cos(θ)
195:
196:            movem.w XL(a6),d0-d1 *d0,d1=パターンの縦横の大きさ
197:
198:            move.w  d0,a5
```


▶ジョン・レノンのイマジンを聞きながら西川善司氏のダッシュ野郎の記事の最後「君のことは忘れない」を読んだとき，なんだかジ～ンときた。P.S.DEFEAT2の追加面を楽しみにしています。
酒本　一幸(18)東京都

```
199:
200:         add.w   d0,d0           *x方向の長さを1.5倍する
201:         add.w   a5,d0           *
202:         lsr.w   #1,d0           *
203:
204:         move.w  d0,d2
205:         move.w  d1,d3
206:
207:         lsr.w   #1,d0           *(d0,d1)=回転の中心
208:         lsr.w   #1,d1           *
209:
210:         move.w  d0,a2           *(a2,a3)=回転の中心
211:         move.w  d1,a3           *
212:
213:         sub.w   d0,d2           *(d0,d1)=パターン中心を
214:         neg.w   d0              *  原点とする左上隅の座標
215:         neg.w   d1              *(d2,d1)=パターン中心を
216:                                 *  原点とする右上隅の座標
217:
218:         move.w  d1,a4
219:         ROT     d0,d1           *左上の点を回転する
220:
221:         exg.l   d1,a4           *(d2,d1)=パターン中心を
222:                                 *  原点とする右上隅の座標
223:         ROT     d2,d1           *右上の点を回転する
224:
225:         move.w  a5,d4           *左下の点を回転する
226:         sub.w   d2,d4           *
227:         move.w  d3,d5           *
228:         sub.w   d1,d5           *
229:
230:         move.w  d1,d3
231:         move.w  a4,d1
232:
233: *       左上    (d0,d1)
234: *       右上    (d2,d3)
235: *       左下    (d4,d5)
236:
237:         movea.l PAT(a6),a1      *(d0,d1)に対応する
238:         adda.w  d0,a1           *  パターン上の位置を
239:         adda.w  d0,a1           *  求める
240:         move.w  a5,d6           *
241:         addq.w  #1,d6           *
242:         add.w   d6,d6           *
243:         add.w   d6,d6           *
244:         muls.w  d1,d6           *
245:         adda.l  d6,a1           *
246:         rts
247:
248: *
249: *       パターン上の線分(d0,d1)-(d2,d3)に沿って
250: *       ポインタを移動する相対アドレスの表を作成する
251: *
252: line_comp:
253:         sub.w   d0,d2
254:         move.w  d2,d4
255:         ABS     d2
256:         SGN     d4
257:
258:         sub.w   d1,d3
259:         move.w  d3,d5
260:         ABS     d3
261:         SGN     d5
262:
263:         add.w   d4,d4
264:         move.w  XL(a6),d0
265:         addq.w  #1,d0
266:         add.w   d0,d0
267:         add.w   d0,d0
268:         muls.w  d0,d5
269:
270:         cmp.w   d3,d2
271:         bcs     yline
272:
273: xline:  move.w  d2,d0   *傾きが緩やかな線分の場合
274:
275:         move.w  d2,d1           *Bresenhamのアルゴリズムに
276:         neg.w   d1              *  必要なパラメータの計算
277:         move.w  d2,d6           *
278:         add.w   d2,d2           *
279:         add.w   d3,d3           *
280:
281: xline0: move.w  d4,d7           *アドレスの相対的な移動量を
282:         add.w   d3,d1           *
283:         bmi     xline1          *
284:         add.w   d5,d7           *
285:         sub.w   d2,d1           *
286: xline1: move.w  d7,(a5)+        *  テーブルに登録していく
287:         dbra    d6,xline0
288:         rts
289:
290: yline:  move.w  d3,d0   *傾きが急な線分の場合
291:
292:         move.w  d3,d1           *Bresenhamのアルゴリズムに
293:         neg.w   d1              *  必要なパラメータの計算
294:         move.w  d3,d6           *
295:         add.w   d2,d2           *
296:         add.w   d3,d3           *
297:
298: yline0: move.w  d5,d7           *アドレスの相対的な移動量を
299:         add.w   d2,d1           *
300:         bmi     yline1          *
301:         add.w   d4,d7           *
302:         sub.w   d3,d1           *
303: yline1: move.w  d7,(a5)+        *  テーブルに登録していく
304:         dbra    d6,yline0
305:         rts
306:
307: *
308: *       クリッピングする
309: *
310:         .offset 0
311: *                                       MINX
312: Min:    .ds.w   1       *MINX           MINY
313:         .ds.w   1       *MINY           MAXX
314: Max:    .ds.w   1       *MAXX           MAXY
315:         .ds.w   1       *MAXY           ----
316: *
317:         .offset 0
318: *
319: D0SAV:  .ds.w   1
320: D2SAV:  .ds.w   1
321: NPIX:   .ds.w   1
322: *
323:         .text
324: *
325: clip:
326:         MINMAX  d1,d2           *d1≦d2を保証する
327:
328:         addi.w  #$8000,d1       *ゲタを履かせる
329:         addi.w  #$8000,d2       *
330:         moveq.l #0,d3
331:         movem.w d0/d2-d3,-(sp)
332: minclip:                        *左端/上端でクリップ
333:         move.w  Min(a0),d6
334:         cmp.w   d6,d1
335:         bcc     maxclip         *ウィンドウ内だった
336:
337:         cmp.w   d6,d2
338:         bcc     min0
339:         bne     outofscrn       *完全にウィンドウ外
340:
341:         move.w  d2,d1
342:         bra     maxclip
343:
344: min0:   moveq.l #0,d3
345: minlp:  move.w  d1,d7
346:         add.w   d2,d7
347:         roxr.w  #1,d7
348:         cmp.w   d6,d7
349:         beq     min2
350:         bcs     min1
351:
352:         move.w  d7,d2
353:         add.w   d3,d0
354:         lsr.w   #1,d0
355:         bra     minlp
356:
357: min1:   move.w  d7,d1
358:         add.w   d0,d3
359:         lsr.w   #1,d3
360:         bra     minlp
361:
362: min2:   move.w  d7,d1
363:         add.w   d3,d0
364:         lsr.w   #1,d0           *d0=ウィンドウ外にはみ出す
365:                                 *  パターンのピクセル数
366:         sub.w   d0,NPIX(sp)
367:
368:         subq.w  #1,d0
369:         bcs     min3
370: skiplp: adda.w  (a4)+,a1        *切り捨てた分
371:         dbra    d0,skiplp       *  参照開始位置を
372:                                 *  ずらす
373: min3:   move.w  D0SAV(sp),d0
374:         move.w  D2SAV(sp),d2
375:
376: maxclip:                        *右端/下端でクリップ
377:         move.w  Max(a0),d6
378:         cmp.w   d6,d2
379:         bls     clipped         *ウィンドウ内だった
380:
381:         cmp.w   d6,d1
382:         bls     max0
383:         bne     outofscrn       *完全にウィンドウ外
384:
385:         moveq.l #0,d0
386:         move.w  d1,d2
387:         bra     clipped
388:
389: max0:   move.w  d1,d4
390:         moveq.l #0,d3
391: maxlp:  move.w  d4,d7
392:         add.w   d2,d7
393:         roxr.w  #1,d7
394:         cmp.w   d6,d7
395:         beq     max2
396:         bcs     max1
397:
398:         move.w  d7,d2
399:         add.w   d3,d0
400:         lsr.w   #1,d0
```

▶きまぐれで食塩水の電気分解をやった。ところが高電圧をかけすぎたり電極をいい加減に選んだりしたため机の上を汚染してしまった。皆さん,化学実験は慎重にしましょう。
宮原　大(18)長野県

```
401:         bra     maxlp
402:
403: max1:   move.w  d7,d4
404:         add.w   d0,d3
405:         lsr.w   #1,d3
406:         bra     maxlp
407:
408: max2:   move.w  d7,d2
409:         add.w   d3,d0
410:         lsr.w   #1,d0
411:
412: clipped:                        *クリッピング完了
413:         subi.w  #$8000,d1       *ゲタを脱がせる
414:         subi.w  #$8000,d2       *
415:
416:         addq.l  #4,sp
417:         add.w   (sp)+,d0        *d0=パターン上の
418:                                 *  点を拾う斜め線分長
419:         rts                     *N=0
420:
421: outofscrn:
422:         addq.l  #6,sp
423:         moveq.l #-1,d0          *N=1
424:         rts
425:
426: *
427: *       斜めに切り出したパターンを
428: *       描画先の大きさに合わせて拡大/縮小するための
429: *       テーブルを作成する
430: *
431: expand_comp:
432:         move.w  d1,(a5)+        *クリッピング後の
433:                                 *  描画先左上隅x/y座標
434:         sub.w   d1,d2
435:         move.w  d2,(a5)+        *描画幅/高さ
436:         beq     exdone
437:
438:         move.w  d2,d1           *Bresenhamのアルゴリズムに
439:         neg.w   d1              *  必要なパラメータの計算
440:         move.w  d2,d3           *
441:         add.w   d2,d2           *
442:         add.w   d0,d0           *
443:
444: explp1: moveq.l #0,d4           *アドレスの相対的な移動量を
445:         add.w   d0,d1           *
446:         bmi     expnxt          *
447: explp2: add.w   (a4)+,d4        *
448:         sub.w   d2,d1           *
449:         bpl     explp2          *
450: expnxt: move.w  d4,(a5)+        *  テーブルに登録していく
451:         dbra    d3,explp1
452:
453: exdone: rts
454:
455:         .end
```

リスト6

```
 1: *       grputのテスト用プログラム
 2: *
 3:         .include        doscall.mac
 4:         .include        iocscall.mac
 5: *
 6:         .xref   grput
 7:         .xref   setcliprect
 8: *
 9:         .text
10:         .even
11: *
12: ent:
13:         lea.l   inisp,sp
14:
15:         clr.l   -(sp)
16:         DOS     _SUPER
17:
18:         lea.l   argbf2,a1
19:         IOCS    _GETGRM
20:
21: *       pea     window
22: *       jsr     setcliprect
23: *       addq.l  #4,sp
24:
25: *       moveq.l #12,d1  *512x512,65536
26:         moveq.l #14,d1  *256x256,65536
27:         IOCS    _CRTMOD
28:         IOCS    _G_CLR_ON
29:
30:         moveq.l #1,d1
31:
32:         lea.l   argbuf,a1
33:         pea.l   (a1)
34: loop:   jsr     grput
35:
36:         add.w   d1,8(a1)                *DEG++
37:
38:         DOS     _KEYSNS
39:         tst.w   d0
40:         beq     loop
41:         DOS     _INKEY
42:         cmpi.b  #$20,d0
43:         bne     done
44:
45: pause:  neg.w   d1
46:         DOS     _INKEY
47:         cmpi.b  #$20,d0
48:         beq     loop
49:
50: done:   move.l  #$0010_0000,-(sp)
51:         DOS     _CONCTRL
52:         DOS     _EXIT
53: *
54:         .data
55:         .even
56: *
57: argbuf:                                 *grputの引数
58: *       .dc.w   96,96,159,159           *縮小テスト
59:         .dc.w   64,64,191,191           *等倍
60: *       .dc.w   32,32,223,223           *拡大テスト
61:         .dc.w   0
62:         .dc.l   pat+512*64+64*2
63:         .dc.w   128-1,128-1
64:         .dc.l   temp
65: *
66: argbf2: .dc.w   128,128,383,383
67:         .dc.l   pat
68:         .dc.l   pate
69: *
70: window: .dc.w   80,80,175,175
71: *
72:         .bss
73:         .even
74: *
75: pat:    .ds.w   256*256                 *描画パターン
76: pate:
77: temp:   .ds.b   4096                    *grputのワーク
78: *
79:         .stack
80:         .even
81: *
82:         .ds.l   4096
83: inisp:
84:         .end    ent
```

リスト7

```
 1: *
 2: *       水平1ライン分描画する専用ルーチンを生成する
 3: *
 4: expand_comp2:
 5:         move.w  d1,(a5)+        *クリッピング後の
 6:                                 *  描画先左上隅x/y座標
 7:         sub.w   d1,d2
 8:         move.w  d2,(a5)+        *描画幅/高さ
 9:         beq     Exdone
10:
11:         move.w  d2,d1           *Bresenhamのアルゴリズムに
12:         neg.w   d1              *  必要なパラメータの計算
13:         move.w  d2,d3           *
14:         add.w   d2,d2           *
15:         add.w   d0,d0           *
16:
17: Explp1: moveq.l #0,d4           *アドレスの相対的な移動量を
18:         add.w   d0,d1           *  求める
19:         bmi     Expnxt          *
20: Explp2: add.w   (a4)+,d4        *
21:         sub.w   d2,d1           *
22:         bpl     Explp2          *
23: Expnxt: move.l  pset(pc),(a5)+  *move,leaの2命令を書き込む
24:         move.w  d4,(a5)+        *アドレスの変位を書き込む
25:                                 *(leaのディスプレースメント)
26: Expskp: dbra    d3,Explp1
27:         move.w  term(pc),(a5)+  *rts命令を書き込む
28:         rts
29:
30: Exdone:                         *1ピクセルしかない場合
31:         move.w  pset(pc),(a5)+  *move命令を書き込む
32:         move.w  term(pc),(a5)+  *rts
33:         rts
34:
35: pset:   move.w  (a4),(a0)+      *1ピクセル書き込む命令
36:         lea.l   0(a4),a4        *参照位置を移動する命令
37: term:   rts                     *最後に置くrts命令
```

マシン語カクテル in Z80's Bar

第24回――もうどうにも止まらない――

暑い夏が到来して，マシン語カクテルに集まって涼んでいる人々。で，話題は何になるかというと，まだまだデバッガにこだわっているようです。前回まで改良を重ねてきたデバッガが，今回はシンボリックデバッガへと変身します。

シナリオ：金子俊一
特別監修：浦川博之

♪カラン，コロ～ン

源光（以下光）：ごめんください。

ようこ（以下Yo）：“ごめん”は置いてませんよ。

マスター（以下M）：ようこちゃんもカビ臭いボケをかましますね。

長老（以下老）：カビ臭いとはなんじゃ。ワシが教えたんじゃぞ。

光&M：やっぱり。

Yo：なあんだ。今いちばんトレンディな小噺だっていうから。

光：“トレンディな小噺”っていうのが泣かせますよね。

老：面白いじゃろ。

光：ちっとも。

老：ワシにケンカ売っとるのか，おぬし。

M：ケンカは売ってませんよ。はい，メニュー。

老：どれどれ，それではワシは熱血飲料でももらおうかの。

光：長老は鉄骨飲料のほうが似合ってるんじゃないかな。

ニシボリック・デバッガ?

Yo：ねえ光君，ニシボリック・デバッガってなあに？

光：ニシボリック・デバッガ？　ニシボリック・サスペンションなら知ってるんだけどなあ。

M：ジェミニのサスですね。

光：マスターも詳しいですね，スミに置けないなあ。

Yo：そんなんじゃなくって，リーボック・デバッガじゃなくって……。

光&M：リーボック・スカイライン？

老：ひょっとして，シンボリックデバッガのことかのう？

Yo：そうそう，それそれ。

M：今日は強烈なボケばっかりぶちかましますね。

光：それも長老に教わったんでしょう。

老：ワシは教えておらんぞ。

Yo：それはともかく，どんなものなの？

光：えっと，X68000ではDB.Xがそうだし，PC-9801なんかではSYMDEB.EXEなんかがそうだね。

Yo：ZAIDや光君が作ったデバッガとかとは違うの？

光：逆アセンブルのときは違うね。

老：ラインアセンブラも違うかのう。

光：でも，ラインアセンブルできる8ビット用のデバッガなんてそうはないですからね。

老：うむ。メモリの都合上かなり苦しいからのう。

光：よく笑い話でありますよね。超強力エディタ&アセンブラ&リンカ&デバッガをひとつのプログラムにすると便利だって。

老：そのかわりフリーエリアが2Kバイトとかじゃろう。

Yo：なんだか話がそれてるわ。

光：シンボリックデバッガは逆アセンブルだけ違うと思ってください。

Yo：それって便利なの？

M：実際に試したほうが早いんじゃないの，光君。

光：その最後の光君ってのは私になにか頼みごとでもあるっていうんじゃ。

Yo：お願い，作って。

光：しょうがないなあ，今夜はシンボリックデバッガでも作りましょう。

最初が肝心肝太郎

光：どうしようかなあ。

Yo：なにが？

光：またパッチ当てをして，プログラムを拡張するのはいやだなあって。

老：ほっほっほ。しょうがないじゃろ，最初から本当にいきあたりばったりで作っておったのじゃから。

光：じゃあ，それはしょうがないとしましょう。

Yo：ほかになにかあるの？

光：シンボルを表示するためにシンボルテーブルを用意するんですけどね，どんなものにしようかなって。

Yo：シンボルってなあに？

光：なんだか質問責めだな。

老：ラベルのことだと思っていれば間違いないじゃろ。

Yo：じゃあ，シンボリックデバッガっていうのはラベルを表示してくれるものなのね。

光：ここですんなりわかっちゃったら，あとのページが埋まんないじゃない。

M：ともかく作ってください。

光：あのね，シンボルを表示するときにデータを探すでしょ。

Yo：サーチするのね。

光：でも，シンボルが多いほどサーチに時間がかかるし，表示時間にもばらつきが出そうでしょ。

Yo：なんとなくわかるわ。だったら最初にソートしておけばいいじゃない。

光：それも考えたけど，データがばらついてないから，あまり解決にはならない。

Yo：どうして？

光：たとえば，5000番地から5FFF番地のプログラムを作ったとしますよね。そしたら，ラベルは圧倒的に5xxx番地のが多くて，ソートしてもあまり意味がなさそうでしょ。

M：実際に調べてみたのかな，光君。

光：ええ，表1を見てください。これは上位1バイトの数を数えたんですけどね。何も書いてないところはデータがなかったところ，書いてあるのはその個数なんです。

M：こりゃあ見事に5000番地台に重なってますね。

光：そうなんですよ。

老：1Fxxや20xxあたりはS-OSのルーチンじゃな。

光：そうなんですよ。

Yo：これは光君が作ったデバッガのラベルを数えたのね。

光：そうなんですよ。

Yo：そればっかり。ほら，山田君がやったハッシュポテトを使ったら。

光：ハッシュ関数ですか。ちょっと面倒臭いかなって思って。

老：若いうちに面倒臭がってばかりいるとロクな大人にならんぞ。

M：ここにいい見本がいるでしょ。

目のつけどころが光でしょ

光：そうだ，もっと簡単にできるかも。

Yo：なにが？

光：ハッシュ関数よりも異常に簡単で，しかもちゃんとばらつきが出そうな方法を思いついたんだよ。

Yo：そんなにすごいの？

光：ハッシュ関数ほどのばらつきにはならないかもしれないけどね。表2を見てください。

老：まあまあバラついておるのう。どうやったのかな，これは。

光：下位の1バイトに注目したんですよ。

老：ほほう。

光：上位1バイトだと，プログラムの位置によってのばらつきはほとんどないけど，下位の1バイトなら完全にランダムなアドレスに割り振られるでしょ。

老：なるほど。

光：表2を数えてみたんですけどね，365個のラベルがあって，1回の検索で22%のラベルが発見できて，同様に2回で54%，3回で84%，4回で95%，5回で100%になるんですよ。

老：ほほう。テストは4月号のプログラムと5月号のプログラムを使ったのじゃな。

光：ええ，そうですが。

老：それではそのラベルのうち20個程度はS-OSで重なっておるじゃろうから，実際はもう少しいい値になりそうじゃのう。

光：おそらくハッシュ関数でやったとしてもたいして変わらない数値になるんじゃないかな。

老：うむ，1～2回くらいは検索が少なくてすむかもしれんがのう。

光：これはいけますね。さっそく，プログラムを作りましょう。

できたけどZEDA用

光：カチャカチャ……。できたっと。

Yo：見せて見せて。

表1 上位1バイトに注目

	$x0	$x1	$x2	$x3	$x4	$x5	$x6	$x7	$x8	$x9	$xA	$xB	$xC	$xD	$xE	$xF
$0x																
$1x																63
$2x	14															
$3x																
$4x																
$5x	12	19	11	13	21	11	21	02	36	35	23	21	18	22	09	14
$6x																
$7x																
$8x																
$9x																
$Ax																
$Bx																
$Cx																
$Dx																
$Ex																
$Fx																

表2 下位1バイトに注目

	$x0	$x1	$x2	$x3	$x4	$x5	$x6	$x7	$x8	$x9	$xA	$xB	$xC	$xD	$xE	$xF
$0x	03	02			01	01	03			02	01	01	02	02	01	02
$1x	03			01	01	02	01	01	03	03		03	02	01	02	03
$2x		05	01	01	05	01		02		01	01		01	01	02	02
$3x	01		01	03		03	01		03			02		02	01	02
$4x			01	02				01		02	02	01	01	02		
$5x	01	01	01	03	02	02			01				01	04	01	
$6x	01	01	02	02		02	02	02	01	01		04	03		02	02
$7x	04	03	04	02	01	03	02	01	01	02	03	03	04	02		01
$8x			01	01	03	02	02	02			01	01		03	03	01
$9x	01	02	01			02	01		03					01	02	01
$Ax	01	01	02	02	01	02	04		01	03	01		02	02	01	02
$Bx	01		03		01	03	03	02	03	01	01	01		01	03	
$Cx		02		02	03	02	01	04	01	01			03	03	01	01
$Dx		04		03	04		03		02	03		01		01	02	04
$Ex		01	02	02	01	02	01	01	01	01	02	02	02	03	02	01
$Fx		05		01	03	01	01	03	02		03		02	01	02	05

05＊4　04＊10　03＊36
02＊59　01＊79　00＊68

表1，表2とも同一プログラム，365ラベルでデータを収集。

コマンドの追加

●S

S-OS用の特殊ワークエリアからシンボルテーブルを展開します。このリストではZEDAで動作するようになっています。ほかのアセンブラでは自作する必要があるかもしれません。

●$

シンボルテーブルをクリアします。この状態で逆アセンブルすると，いままでと同じ出力になります。

●P

シンボルテーブルのアドレスを表示します。

光：えっと，まずアセンブラのワークエリアからラベルをシンボルテーブルに振り分けなくちゃ。

Yo：新しいコマンドが増えているのね。

光：そこで逆アセンブルっと。

Yo：すごい，なんだかわかりやすくなったわね。

光：そのためのシンボリックデバッガですからね。

M：見比べるとよくわかりますね。

光：それではサンプルを出しておきますか。リスト1がソースリスト，リスト2は普通の逆アセンブル，リスト3が今回のシンボリックデバッガの逆アセンブルです。

老：これはなかなかじゃのう。

M：スピードも心配してたほどは遅くなりませんでしたね。

Yo：よかったよかった。

光：ただし，この中のシンボルテーブルを作るルーチンはZEDA用だから，ほかのアセンブラだと動かないかもしれませんよ。

老：どうすればいいのじゃ？

光：やっぱり自分で作ってもらうしかないでしょうね。

Yo：何か資料はないの？

光：それじゃあ作り方の資料もサービスしておきましょう。

Yo：ほかのアセンブラで動くやつを作ったら送ってね。

老：誰に向かっていっておるのじゃ？

Yo：ただのひとり言です。

光：そうそう，6400番地からシンボルテーブルを作るから，あまりにいっぱいのラベルを扱うときはアドレスに注意してね。300ラベル程度なら7000番地には届かないだろうから大丈夫だと思うけど。

Yo：300ラベルっていったらかなり長いプ

シンボルテーブルの作り方

```
ABC  EQU  $1234
DEF  EQU  $3456
GHI  EQU  $7834
```

というラベルがあったとしよう。

まず，ABCの下位バイト$34に注目する。下位バイトのテーブル，SIMO2＋$34×2番地に上位バイトが格納されるアドレスを入れる。上位バイトは，

$12, A, B, C, $0D, $00, $00

を1セットとして，

SIMO2＋$56×2番地に次のデータが入っているアドレスを入れる。

$34, D, E, F, $0D, $00, $00

これで2つめが終わり。

最初の1バイトが上位バイト，そこからラベル名，$0Dでラベル名終わり。その次の2バイトが次のラベルのポインタになっている。下位バイトテーブルと上位バイトポインタは＄0000が入っているときは“nul”だと思ってくれればわかりやすいかもしれない。

さて，この次のGHIであるが，ABCと下位バイトが同じため，SIMO2＋$34×2番地にはすでにABCのアドレスが入っている。そこで，ABCのポインタにGHIのデータが入っているアドレスを書き込む。

$12, A, B, C, $0D, $XY, $ZW

ZWXY番地：

$78, G, H, I, $0D, $00, $00

となる。

プログラムの仕様上，＄0000番地にはシンボルを与えることができない。

これでわからないときはプログラムを解析しよう。MKTBLというラベルからがそのルーチン。ちなみにZEDAのラベルテーブルでは，

ラベル名，$0D，アドレス，ラベル名，$0D，アドレス，………，ラベル名，$0D，アドレス，$00

となっていて，最後の$00がエンドコード。

リスト1

```
0000                1 ;
0000                2 ;        Sample Program for Symbolic Debugger
0000                3 ;
0000                4
8000                5          ORG     $8000
8000                6
8000                7 #PRTHL   EQU     $1FBE
8000                8 #MPRINT  EQU     $1FE2
8000                9 #LTNL    EQU     $1FEE
8000               10 #PRINT   EQU     $1FF4
8000               11 #VER     EQU     $1FF7
8000               12
8000               13 START
8000 C3 06 80      14          JP      TEST
8003 C3 15 80      15          JP      TEST2
8006               16 TEST
8006 3E 24         17          LD      A,"$"     ; S-OS Version check
8008 CD F4 1F      18          CALL    #PRINT
800B CD F7 1F      19          CALL    #VER
800E CD BE 1F      20          CALL    #PRTHL
8011 CD EE 1F      21          CALL    #LTNL
8014 C9            22          RET
8015               23 TEST2
8015 AF            24          XOR     A         ; Memory clear
8016 32 25 80      25          LD      (@WORK),A
8019               26          ;
8019 21 25 80      27          LD      HL,@WORK
801C 11 26 80      28          LD      DE,@WORK+1
801F 01 DA 0F      29          LD      BC,$9000-@WORK-1
8022 ED B0         30          LDIR
8024 C9            31          RET
8025               32 @WORK
8025 00            33          DS      1
OBJECT CODE END 8025
```

リスト2

```
8000 C3 06 80      JP      $8006
8003 C3 15 80      JP      $8015
8006 3E 24         LD      A,$24
8008 CD F4 1F      CALL    $1FF4
800B CD F7 1F      CALL    $1FF7
800E CD BE 1F      CALL    $1FBE
8011 CD EE 1F      CALL    $1FEE
8014 C9            RET
8015 AF            XOR     A
8016 32 25 80      LD      ($8025),A
8019 21 25 80      LD      HL,$8025
801C 11 26 80      LD      DE,$8026
801F 01 DA 0F      LD      BC,$0FDA
8022 ED B0         LDIR
8024 C9            RET
8025 00            NOP
```

リスト3

```
:START
8000 C3 06 80      JP      $8006:TEST
8003 C3 15 80      JP      $8015:TEST2
:TEST
8006 3E 24         LD      A,$24
8008 CD F4 1F      CALL    $1FF4:#PRINT
800B CD F7 1F      CALL    $1FF7:#VER
800E CD BE 1F      CALL    $1FBE:#PRTHL
8011 CD EE 1F      CALL    $1FEE:#LTNL
8014 C9            RET
:TEST2
8015 AF            XOR     A
8016 32 25 80      LD      ($8025:@WORK),A
8019 21 25 80      LD      HL,$8025:@WORK
801C 11 26 80      LD      DE,$8026
801F 01 DA 0F      LD      BC,$0FDA
8022 ED B0         LDIR
8024 C9            RET
:@WORK
8025 00            NOP
```

ログラムじゃない。それだけで300行もあるんだもの。

光：そうだね。よっぽどのことがないかぎりは心配しなくていいと思うけどね。

老：プログラムは短いのじゃが，ワークエリアが大きいから6FFF番地まで使うと考えたほうがよさそうじゃの。

光：そうですね。

Yo：ねえ，光君。画面にいっぱい出てくる「＊」はなんなの？

光：ラベルの数です。実際はラベルの数＋1個の「＊」が表示されますけどね。ラベルの個数の目安にしてみてください。

Yo：なるほどね，星が4行も5行もあるようだと，7000番地を超えているかもしれないってわけね。

光：そういうこと。まあ具体的なアドレスは「P」コマンドで見れますけど。

Yo：お疲れさまでした。

光：では，私は夏休みのレポートがあるので，このへんで失礼します。

Yo：すっかり夏よねえ。

老：ミーン，ミンミンミンミーン。

―つづく―

リスト4

```
0000           1 ;       Symbolic Debugger
0000           2 ;
0000           3 ;          by Hikaru Minamoto
0000           4
0000           5          OFFSET  $A000-$6000
6000           6          ORG     $6000
6000           7
6000           8 ;Label          Address   Break
6000           9
6000          10 #PEEK    EQU     $1F94    ; AF
6000          11 #PRTHL   EQU     $1FBE    ; AF
6000          12 #MPRINT  EQU     $1FE2    ; AF,DE
6000          13 #MSG     EQU     $1FE8    ; F
6000          14 #LTNL    EQU     $1FEE    ; nothing
6000          15 #PRINT   EQU     $1FF4    ; F
6000          16 #VER     EQU     $1FF7    ; HL
6000          17
6000          18 COLD     EQU     $5000
6000          19 COM      EQU     $5099
6000          20 COM2     EQU     $509D
6000          21 CLOSE    EQU     $50DD
6000          22 ADR      EQU     $56B1
6000          23 NN       EQU     $5843
6000          24 RJ       EQU     $5854
6000          25
6000          26
6000          27 PATCH
6000 3E C3    28          LD      A,$C3            ; $C3=JP
6002          29          ;
6002 21 00 60 30          LD      HL,PATCH
6005 22 01 50 31          LD      (COLD+1),HL
6008          32          ;
6008 21 56 60 33          LD      HL,NN_SYM
600B 32 43 58 34          LD      (NN),A
600E 22 44 58 35          LD      (NN+1),HL
6011          36          ;
6011 21 61 60 37          LD      HL,RJ_SYM
6014 32 54 58 38          LD      (RJ),A
6017 22 55 58 39          LD      (RJ+1),HL
601A          40          ;
601A 21 48 60 41          LD      HL,ADR_SYM
601D 22 B1 56 42          LD      (ADR),HL
6020          43          ;
6020 21 31 60 44          LD      HL,COMM
6023 32 99 50 45          LD      (COM),A
6026 22 9A 50 46          LD      (COM+1),HL
6029 AF       47          XOR     A                ; $00=NOP
602A 32 9C 50 48          LD      (COM+3),A
602D          49          ;
602D CD F7 1F 50          CALL    #VER
6030 C9       51          RET
6031          52
6031          53 COMM
6031 FE 21 CA 54          IF A="!" JP CLOSE
6034 DD 50
6036 FE 53 CA 55          IF A="S" JP MKTBL
6039 D1 60
603B FE 24 CA 56          IF A="$" JP CLRTBL
603E A4 60
6040 FE 50 CA 57          IF A="P" JP ADR_PRT
6043 5B 61
6045 C3 9D 50 58          JP      COM2
6048          59
6048          60 ADR_SYM
6048 E5       61          PUSH    HL
6049 CD 73 60 62          CALL    SYM_SER
604C E1       63          POP     HL
604D 30 03    64          JR      NC,ADR_SYM2
604F CD EE 1F 65          CALL    #LTNL
6052          66 ADR_SYM2
6052 CD BE 1F 67          CALL    #PRTHL
6055 C9       68          RET
6056          69
6056          70 NN_SYM
6056 60 69    71          LD      HL,BC
6058 23       72          INC     HL
6059 4E       73          LD      C,(HL)
605A 23       74          INC     HL
605B 46       75          LD      B,(HL)
605C 60 69    76          LD      HL,BC
605E C3 6B 60 77          JP      SYM_PRT
6061          78 RJ_SYM
6061 03       79          INC     BC
6062 0A       80          LD      A,(BC)
6063 03       81          INC     BC
6064 6F       82          LD      L,A
6065 07       83          RLCA
6066 3E 00    84          LD      A,0      ; NOT (XOR A)
6068 9F       85          SBC     A,A
6069 67       86          LD      H,A      ; H=00orFF
606A 09       87          ADD     HL,BC
606B          88 SYM_PRT
606B 3E 24    89          LD      A,"$"
606D CD F4 1F 90          CALL    #PRINT
6070 CD BE 1F 91          CALL    #PRTHL
6073          92          ;
6073          93 SYM_SER                   ; SEARCH
6073 44       94          LD      B,H
6074          95          ;
6074 5D       96          LD      E,L
6075 16 00    97          LD      D,$00
6077 21 00 62 98          LD      HL,SIMO2
607A 19       99          ADD     HL,DE
607B 19      100          ADD     HL,DE
607C         101          ;
607C 7E      102          LD      A,(HL)
607D 5F      103          LD      E,A
607E 23      104          INC     HL
607F 56      105          LD      D,(HL)
6080 B2      106          OR      D
6081 C8      107          RET     Z        ; Not find symbol
6082         108          ;
6082 60      109          LD      H,B
6083         110 TOP
6083 1A      111          LD      A,(DE)
6084 BC      112          CP      H
6085 28 12   113          JR      Z,PR_SYM
6087         114 SYM_LP
6087 13      115          INC     DE
6088 1A      116          LD      A,(DE)
6089 FE 0D   117          CP      $0D
608B 20 FA   118          JR      NZ,SYM_LP
608D         119          ;
608D 13      120          INC     DE
608E 1A      121          LD      A,(DE)
608F 4F      122          LD      C,A
6090 13      123          INC     DE
6091 1A      124          LD      A,(DE)
6092 57      125          LD      D,A
6093 59      126          LD      E,C
6094 B3      127          OR      E
6095 C8      128          RET     Z        ; Not find symbol
6096 C3 83 60 129         JP      TOP
6099         130          ;
6099         131 PR_SYM
6099 3E 3A   132          LD      A,":"
609B CD F4 1F 133         CALL    #PRINT
609E 13      134          INC     DE
609F CD E8 1F 135         CALL    #MSG     ; Print SYMBOL
60A2 37      136          SCF
60A3 C9      137          RET
60A4         138
60A4         139 CLRTBL
60A4 CD E2 1F 140         CALL    #MPRINT
60A7 20 53 79 141         DM      " Symbol table clear."
60AA 6D 62 6F
60AD 6C 20 74
60B0 61 62 6C
60B3 65 20 63
60B6 6C 65 61
60B9 72 2E
60BB 0D 00   142          DB      $0D,$00
```

```
60BD                143        ;
60BD 21 00 62       144        LD      HL,SIMO2
60C0 11 01 62       145        LD      DE,SIMO2+1
60C3 01 FF 01       146        LD      BC,256*2-1
60C6 36 00          147        LD      (HL),$00
60C8 ED B0          148        LDIR
60CA 21 00 64       149        LD      HL,KAMI2
60CD 22 9D 61       150        LD      (@KAMI),HL
60D0 C9             151        RET
60D1                152
60D1                153 ;
60D1                154 ;      Make Symbol Table  . for ZEDA-3
60D1                155 ;
60D1                156
60D1                157 MKTBL
60D1 CD A4 60       158        CALL    CLRTBL
60D4 CD E2 1F       159        CALL    #MPRINT
60D7 20 4E 6F       160        DM      " Now, making symbols."
60DA 77 2C 20
60DD 6D 61 6B
60E0 69 6E 67
60E3 20 73 79
60E6 6D 62 6F
60E9 6C 73 2E
60EC 0D 00          161        DB      $0D,$00
60EE 21 00 10       162        LD      HL,$1000
60F1                163 MAKE
60F1 3E 2A          164        LD      A,"*"
60F3 CD F4 1F       165        CALL    #PRINT
60F6 CD 94 1F       166        CALL    #PEEK
60F9 B7             167        OR      A
60FA 20 12          168        JR      NZ,MAKE1
60FC CD E2 1F       169        CALL    #MPRINT
60FF 0D             170        DB      $0D
6100 20 43 6F       171        DM      " Completed."
6103 6D 70 6C
6106 65 74 65
6109 64 2E
610B 0D 00          172        DB      $0D,$00
610D C9             173        RET
610E                174        ;
610E                175 MAKE1
610E ED 5B 9D       176        LD      DE,(@KAMI)
6111 61
6112                177        ;
6112                178 GET_LBL
6112 13             179        INC     DE
6113 12             180        LD      (DE),A
6114 FE 0D          181        CP      $0D
6116 28 06          182        JR      Z,MAKE2
6118                183        ;
6118 23             184        INC     HL
6119 CD 94 1F       185        CALL    #PEEK
611C 18 F4          186        JR      GET_LBL
611E                187 MAKE2
611E AF             188        XOR     A
611F 13             189        INC     DE
6120 12             190        LD      (DE),A  ; Next Add.
6121 13             191        INC     DE
6122 12             192        LD      (DE),A  ; Next Add.
6123                193        ;
6123 ED 4B 9D       194        LD      BC,(@KAMI)
6126 61
6127 13             195        INC     DE
6128 ED 53 9D       196        LD      (@KAMI),DE
612B 61
612C                197        ;
612C 23             198        INC     HL
612D 23             199        INC     HL
612E CD 94 1F       200        CALL    #PEEK
6131 02             201        LD      (BC),A
6132                202        ;
6132 2B             203        DEC     HL
6133 CD 94 1F       204        CALL    #PEEK
6136 E5             205        PUSH    HL
6137 21 00 62       206        LD      HL,SIMO2
613A 5F             207        LD      E,A
613B 16 00          208        LD      D,0
613D 19             209        ADD     HL,DE
613E 19             210        ADD     HL,DE
613F                211        ;
613F                212 MAKE25
613F 54 5D          213        LD      DE,HL
6141 7E             214        LD      A,(HL)
6142 23             215        INC     HL
6143 66             216        LD      H,(HL)
6144 6F             217        LD      L,A
6145 B4             218        OR      H
6146 20 0A          219        JR      NZ,MK_LP
6148                220        ;
6148 62 6B          221        LD      HL,DE
614A 71             222        LD      (HL),C
614B 23             223        INC     HL
614C 70             224        LD      (HL),B
614D                225        ;
614D E1             226        POP     HL
614E 23             227        INC     HL
614F 23             228        INC     HL
6150 18 9F          229        JR      MAKE
6152                230 MK_LP
6152 23             231        INC     HL
6153 7E             232        LD      A,(HL)
6154 FE 0D          233        CP      $0D
6156 20 FA          234        JR      NZ,MK_LP
6158 23             235        INC     HL
```

```
6159 18 E4          236        JR      MAKE25
615B                237
615B                238 ADR_PRT
615B CD E2 1F       239        CALL    #MPRINT
615E 53 79 6D       240        DM      "Symbol table address"
6161 62 6F 6C
6164 20 74 61
6167 62 6C 65
616A 20 61 64
616D 64 72 65
6170 73 73
6172 0D             241        DB      $0D
6173 46 72 6F       242        DM      "From :$"
6176 6D 20 3A
6179 24
617A 00             243        DB      $00
617B                244        ;
617B 21 00 64       245        LD      HL,KAMI2
617E CD BE 1F       246        CALL    #PRTHL
6181 CD EE 1F       247        CALL    #LTNL
6184                248        ;
6184 CD E2 1F       249        CALL    #MPRINT
6187 54 6F 20       250        DM      "To   :$"
618A 20 20 3A
618D 24
618E 00             251        DB      $00
618F 21 9D 61       252        LD      HL,@KAMI
6192 7E             253        LD      A,(HL)
6193 23             254        INC     HL
6194 66             255        LD      H,(HL)
6195 6F             256        LD      L,A
6196 CD BE 1F       257        CALL    #PRTHL
6199 CD EE 1F       258        CALL    #LTNL
619C C9             259        RET
619D                260
619D                261 @KAMI
619D 00 64          262        DW      KAMI2
619F 00 00 00       263        DS      $6200-@KAMI-2
61A2 00 00 00
61A5 00 00 00
61A8 00 00 00
61AB 00 00 00
61AE 00 00 00
61B1 00 00 00
61B4 00 00 00
61B7 00 00 00
61BA 00 00 00
61BD 00 00 00
61C0 00 00 00
61C3 00 00 00
61C6 00 00 00
61C9 00 00 00
61CC 00 00 00
61CF 00 00 00
61D2 00 00 00
61D5 00 00 00
61D8 00 00 00
61DB 00 00 00
61DE 00 00 00
61E1 00 00 00
61E4 00 00 00
61E7 00 00 00
61EA 00 00 00
61ED 00 00 00
61F0 00 00 00
61F3 00 00 00
61F6 00 00 00
61F9 00 00 00
61FC 00 00 00
61FF 00
6200                264
6200                265        ORG     $6200
6200                266 SIMO2
6200 00 00 00       267        DS      256*2
6203 00 00 00
6206 00 00 00
6209 00 00 00
620C 00 00 00
620F 00 00 00
        :
        :
63FE 00 00
6400                268 KAMI2
```

X1用アスレチックアクションゲーム

Manual Runner

斬新なアイデアでゲームを作る柴田氏の最新作です。「走る足」の両足の動きをバランスよく制御し、100m競走をしましょう。きっと意識して走ることの難しさを思い知ることになるでしょう。さあ世界新記録を更新できるかな？

Shibata Atsushi
柴田　淳

歩くとき

筒井康隆氏の短編に『歩くとき』というのがあります。この小説，ただ単に歩くときは足をどう動かせばいいのかとか，曲がるときはどうするかといったことを彼一流のスラップスティックな文章を交じえて書いてあるのです。ドタバタの面白さはあるにせよ，彼の寡作が生み出した取るに足らない作品，一見そんな衣をまとった短編です。

ところで，彼はSF作家として世に出てはきたものの，彼の代表的な作品のなかに，頭から足の先までSFといったものはあまり見受けられないような気がします。実際伝え聞くところによると，筒井康隆氏はたいそう機械音痴で，自分で買ったステレオの配線すらできないそうなのです。

そういったことをふまえて，改めて先の短編『歩くとき』を読んでみると，ドタバタに織り混ぜられている徹底してメカニカルで忠実な描写の部分が，なおさら浮き彫りになってきます。股関節右側を制御する筋肉に力を加えだとか，アキレス腱別名下腿三頭筋腱別名踵骨腱にもさらに力を加えだとか。よくもまあと思うくらいに医学用語が飛び出してきて，いちいち読者に歩くときはどこをどんなふうに動かしているのかということを確認させるのです。

人間が歩くときは筋肉をどう動かそうとか考えているわけではなく，たんにどこそこに行こうとか思うだけで，ときには無意識にさえ歩きだしているものです。

その半無意識的な行動を緻密な描写で意識させる作業は，僕にはどことなくプログラミングの作業に似ているように思えてなりません。目的とする操作をいくつもの要素に分けていき，最終的にひとつに統合する。確かに彼は機械音痴なのかもしれませんが，そこはそれ。彼一流の機械音痴らしからぬ直感のようなもので，プログラムを組むときの手法をさぐりあててしまったのではないか。僕は『歩くとき』を読んだとき，そんな気がしてなりませんでした。

少々深読みのしすぎでしょうか。ただし，この短編は10年以上も前，いまのように誰もがパソコンを持っていて，プログラミングを楽しんでいるという状況とはほど遠い時代に書かれたものなのです。

走るとき　1

走る人間をコンピュータのうえで再現するにはどうしたらいいでしょうか。まず誰でも考えるのが，いくつかアニメパターンを用意して，それを順番に表示するという手法でしょう。ゲームなどでも，この手法がもっともよく使われているようです。

しかし，それではあまりにも華がないではありませんか。なにかもっとほかの方法で人間を走らせることはできないか。僕はそんなことを常々考えていたのでした。

プログラムの入力

このプログラムはBASICとマシン語の2つのプログラムから成っています。

まずリスト1をBASICから入力します。間違いがないことを確認したら，“MR.BAS”などのファイル名でセーブします。

BASIC部分の入力が終わったら，次はリスト2のマシン語部分を各種入力ツールやモニタから入力します。CRCチェックバイトを確認し，間違いがなければ，“MR.BIN”のファイル名でセーブしてください。カセットの場合は，リスト2の直後にセーブしてください。

入力がすべて終わったなら，BASICからリスト1を読み込み，走らせればゲームが始まります。

ゲームを始めるとまずタイトルが現れます。ここで任意のキーを押すと，設定画面に移ります。

最初に足の形を設定します。1が正常な足（といっても足だけというのはかなり不気味ですが），2が異常な足です。1を選ぶのが無難だと思いますが，操作に慣れてきたら2の足で走ってみましょう。楽々高タイムを出せるはずです。

次にスピードを設定します。1を押すと表示の間隔が細かくなり，その分スピードが遅くなります。2を押すと間隔が粗くなり，スピードが速くなります。最初は2を選ぶのがいいでしょう。

最後はゲームのモードを設定します。1は10000ドットレース，2は50000ドットレース，3は練習モードです。100ドットが1メートルの換算になっていますので，1を選べば100メートル走となるわけです。3の練習モードについてはあとで説明します。

すべての設定が終わったらゲーム開始です。なお，このゲームをするにあたってはジョイスティックが必要です。ゲームを始める前に，ジョイスティックポートの1に差し込んでおいてください。

操作方法

さて，前にも書いたとおり，このゲームでは少々変わった操作方法を採用しています。人間を走らせるときには，ジョイスティックの2つのボタンしか使いません。かといって，ボタンを速く叩けば叩くほど速く走るというわけでもありません。

まず，基本的なボタンの効用について説明しましょう。

設定が終わると現れる2本の足は，2つのボタンのうち片方を押すことによって，大腿部が前方に持ち上がるようになっています。ですから当然放せば伸びるわけですが，足は伸び切る寸前に膝関節から曲がりだし，その結果前方に進むことができるのです。

この操作を交互に繰り返すことによって，

2本の足はあれよあれよというあいだに走りだすはずなのですが，現実はそう甘くありません。

●練習モード

速く走るためのこつは,2つのボタンを押したり放したりするタイミングにあるのですが，そのタイミングをつかむのが難しいのです。うまくしないと後ろの足の爪先を地面に引っ掛けてしまい，逆に後ろに進んでしまうことすらあります。2，3分，自分でタイミングをつかめるかどうか試してみるのは楽しいかもしれませんが,それも5分も経つといいかげんに飽きてくるでしょう。

そんなときには短気を起こして電源を切ってしまうようなことはせず，モード設定で3の練習モードを選びましょう。

練習モードでは，走るのに最適なボタン操作のタイミングを教えてくれます。PUSHと表示されたらボタンを押し，RELEASEと表示されたらボタンを放します。指示が現れているあいだは入力があるまでゲームが止まっていますので，最初はあせらず，指示に忠実にボタンを押すように心掛けてください。

だいたい要領が飲み込めたら，今度は指示が現れている時間をなるべく少なくするようにボタンを押します。止まっている時間がほとんどなくなり，操作に慣れてきたらエスケープキーを押し普通のモードにもどり，自分の力量を試してみましょう。かなりのタイムアップが望めるはずです。

なお，ゲーム中にエスケープキーを押すと途中でゲームを終えることができます。操作を間違えたときなどに便利でしょう。

ゲームが終わったときスペースキーを押すと前回と同じ状態でゲームを続けることができます。ほかのキーを押すと，タイムレコードが表示され，さらにスペース以外のキーを押すとタイトル画面に戻ります。

また，タイトルが表示されているときにエスケープキーを押すと，プログラムから抜け出します。

走るとき 2

操作方法をお読みになればわかると思いますが，このゲームではかなり遠回りしてコンピュータ上で走る人間を再現しています。プログラムのサイズから考えても，足の形状の数分の座標データを用意したほうが明らかに短くてすむはずです。

しかし，遠回りをすればその分『遊び』が出てくるわけで，事実このゲームではいろいろなことができるようになっています。

その『遊び』の部分はさておき，ここではどのようにして人間を走らせているかを説明することにしましょう。

まずお気づきのとおり，このゲームでは直立した足の座標データだけを用意しておき，それを回転させて曲がった足を表現しています。ただし，この場合は単純に回転させればいいというわけではなく，『関節』という曲者があるのです。

関節の処理で面倒なのが，根本から遠いほうの関節の座標が根本に近い関節の回転に影響される，ということでしょう。つまり，2つの関節を同時に回転させるとき，根本の関節の回転によってもうひとつの関節を回転させる中心座標が変わるだけでなく，回転させる角度まで変わってしまうのです。

この変化を代数的に計算できないこともないでしょうが，マシン語のプログラムでそんなことをするのも面倒です。ここではおそらく関節の処理でいちばんオーソドックスだと思われる方法で切り抜けています。

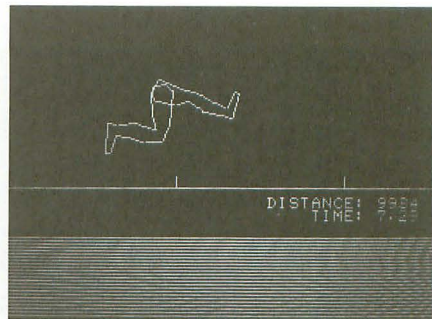
要するに，根本から遠いほうの関節を回転させ，その座標を根本に近いほうの関節の座標を中心に，もう一度回転させます。多関節の場合も，この操作を関節の数だけ繰り返せばいいのです。

さて，ここまでで足の関節を回転させることができるようになったわけですが，ただ回転させるだけでは足がブラブラと揺れているだけにしか見えませんので，ここでもうひと工夫しなければなりません。つまり足の関節の回転に合わせ，飛び上がったり着地したりと臨機応変に座標（具体的には縦座標）を変えなければならないのです。

このプログラムでは片足につき2つの座標，かかとと爪先の座標を参照して，地面に着地した時点でのX，Y成分を割り出し，その分を移動量に加算して走る動作を実現しています。簡単にいってしまえば足が地面にめり込む分，上に上がったり前に進んだりするのです。

ざっといってしまえばこんなものですが，実際これをプログラミングするとなるといろいろな問題点がありました。まあそんなことをいちいち書いても，それを読んでくれるほど読者の皆様も寛大ではないでしょうから，ややこしいことはこのへんでやめておきましょうか。

より速く走るために

このゲームは，やはりなんといっても大勢でわいわい騒ぎながらやるのが楽しいと思います。人間とは本質的にこうなのかも

ひたすら走れ！

しれませんが，どんなささやかなものであれ競争というのは楽しいものです。実質的な利益がかかっていなく，負けても実害がなにもないとなればなおさらそうでしょう。

練習を重ねることによってタイムも上がっていきます。練習モードをマスターすれば15秒を切れるようになっているはずです。しかしそこが第一の壁となるでしょう。

タイムが上がってもせいぜい13秒くらいが限度で，それ以上は困難です。このことの原因はおそらくスタートにあります。スタートさえ工夫すれば，10秒台前半くらいのラップは出せるようになります。

ヒントをいうと，低い位置からスタートを切るよりも，高く飛び上がって蹴りだしたほうが出だしがいいということです。スタートを切ってからすぐにエスケープキーを押し，その後スペースキーを押すと，繰り返してスタートだけを練習することができます。タイミングさえつかめば（結局これにつきますが）スピードの乗ったスタートが切れるようになるでしょう。

それ以上のラップを出すにはどうすればいいでしょうか。まず足を引きつけるのを微妙に早めます。微妙にです。そしてミスをしないこと。完璧な走りが要求されます。ちなみに僕のベストラップは9秒92です。

* * *

このゲーム，結構長くなってしまいました。全部打ち込むとなると骨が折れるでしょうが暇な人は入力してみてください。

寛大にもこの冗長なプログラムを打ち込んでいただいて，なおかつ飽きるまで遊んでくれた心あるX1ユーザーの方には，お礼として暇潰しの技を教えましょう。

ボタンを押して足が曲がりきったところで，トリガーを連射します。するとだんだん足が伸びていくはずです。

両足が完全に伸びたところで，おもむろに片方のボタンを放します。そして，

「どうしてそゆことするのかなっ！」

と奇声を発すれば，欽ちゃん飛びの完成です。あ，くだらなすぎますか。すいません。

リスト1

```
1000 ' MANUAL RUNNER
1010 '
1020 '       (ats)   1991  JUN.
1030 '
1040 CLEAR &HF000:GOSUB "INIT"
1050 GOSUB "OPEN"
1060 GOSUB "SCREEN"
1070 GOSUB "SETTING"
1080 GOSUB "GAME"
1090 GOSUB "RECORD":GOTO 1050
1100 LABEL "GAME"
1110 CLS:LINE(0,0)-(319,121),PSET,0,BF
1120 IF DI%=2 THEN "EXERCISE"
1130 LOCATE  0,16:PRINT TAB(23);"DISTANCE:
1140 LOCATE 27,17:PRINT "TIME:        "
1150 LOCATE 27,10:PRINT "             "
1160 IF DI%=0 THEN CALL &HF000 ELSE CALL &HF006
1170 IF INKEY$(0)=" THEN SOUND 7,255:GOTO 1280
1180 TI%=PEEK(&HFC85)+PEEK(&HFC86)*256
1190 FOR I%=0 TO 9
1200 IF TR%(DI%,SP%,I%)>TI% THEN 1220
1210 NEXT:GOTO 1280
1220 FOR J%=9 TO I%+1 STEP -1
1230 TR%(DI%,SP%,J%)=TR%(DI%,SP%,J%-1):NEXT
1240 TR%(DI%,SP%,I%)=TI%
1250 LOCATE 12, 6:PRINT "YOU ESTABLISHED A NEW RECORD"
1260 A$="TH":IF I%<3 THEN A$=MID$("STNDRD",I%*2+1,2)
1270 LOCATE 19,10:PRINT I%+1;A$+" >"
1280 LOCATE 9,18:PRINT "HIT SPACE TO CONTINUE":KEY 0,""
1290 I$=INKEY$:IF I$=" " THEN "GAME"
1300 IF I$<>"" THEN RETURN ELSE 1290
1310 LABEL"EXERCISE"
1320 CLS:LOCATE 23,17:PRINT "EXERCISE"
1330 LOCATE 0,13:PRINT " "
1340 CALL &HF0BD:GOTO 1170
1350 LABEL"RECORD"
1360 IF DI%=2 THEN RETURN
1370 LOCATE 0 ,1:PRINT "TODAY'S RECORD"
1380 FOR I%=0 TO 9:LOCATE 1,I%+3
1390 PRINT USING "## : ##.##";I%+1,TR%(DI%,SP%,I%)/100
1400 NEXT
1410 I$=INKEY$:IF I$=" " THEN RETURN 1080
1420 IF I$<>"" THEN RETURN ELSE 1410
1430 LABEL "SETTING"
1440 LOCATE 5,1:PRINT "LEG FORM   (1 / 2) ?";
1450 I$=INKEY$(1):IF I$<>"1" AND I$<>"2" THEN 1450
1460 MEM$(&HFC59,36)=LF$(VAL(I$)-1)
1470 LOCATE 5,3:PRINT "GAME SPEED (1 / 2) ?";
1480 I$=INKEY$(1):IF I$<>"1" AND I$<>"2" THEN 1480
1490 SP%=VAL(I$)-1:POKE &HFC3B,SP%+2
1500 LOCATE 5,5:PRINT "GAME MODE  (1 - 3) ?";
1510 I$=INKEY$(1):IF VAL(I$)<1 OR VAL(I$)>3 THEN 1510
1520 DI%=VAL(I$)-1
1530 RETURN
1540 LABEL "INIT"
1550 WIDTH 40:INIT:COLOR 4:CSIZE 0:DIM TR%(1,1,9),LF$(1)
1560 IF PEEK(&HF000)<>33 THEN LOADM "MR.BIN"
1570 SCREEN0,1:GOSUB 1750
1580 FOR I%=0 TO 9:TR%(0,0,I%)=4000:NEXT
1590 FOR I%=0 TO 9:TR%(0,1,I%)=4000:NEXT
1600 FOR I%=0 TO 9:TR%(1,0,I%)=9000:NEXT
1610 FOR I%=0 TO 9:TR%(1,1,I%)=9000:NEXT
1620 LF$(0)=MEM$(&HFB9B,36):LF$(1)=MEM$(&HFBBF,36)
1630 MEM$(&HFD00,40)=STRING$(0,40):RETURN
1640 LABEL"OPEN"
1650 SCREEN1,1:PALET
1660 LOCATE 10,14:PRINT"HIT ANY KEY TO START"
1670 I$=INKEY$:IF I$="" THEN 1670
1680 IF I$=CHR$(&H1B) THEN CLS 4:END
1690 RETURN
1700 LABEL "SCREEN"
1710 SCREEN0,0,3:CLS 4
1720 LINE(0,152)-(319,199),PSET,BF,CHR$(0,255)
1730 LINE(0,122)-(319,122),PSET,1
1740 RETURN
1750 LINE(0,86)-(319,108),PSET,7,BF
1760 RESTORE 1900:FOR I%=0 TO 18
1770 READ A$:LINE(0,88+I%)-(319,88+I%),PSET,BF,HEXCHR$(A$)
1780 NEXT:Y%=87:C%=0:GOSUB 1820
1790 RESTORE 2220:FOR I%=1 TO 5
1800 READ X%:PAINT (X%,95),0:NEXT
1810 Y%=87:C%=7:GOSUB 1820:RETURN
1820 FOR I%=1 TO 12
1830 RESTORE MID$("MANUALRUNNER",I%,1)
1840 X%=I%*26-24-(I%>6)*8+(I%>4)*3+(I%>7)*2:GOSUB 1860
1850 NEXT:RETURN
1860 READ X1%,Y1%
1870 READ X2%,Y2%:IF X1%=99 OR X2%=99 OR X2%=-1 THEN 1890
1880 LINE(X%+X1%,Y%+Y1%)-(X%+X2%,Y%+Y2%),PSET,C%
1890 X1%=X2%:Y1%=Y2%:IF X2%<>-1 THEN 1870 ELSE RETURN
1900 DATA FF0022,FF0055,FF00B6,FF00DD,FF007F,FF08FF,FF22FF
1910 DATA FF88FF,FF22FF,FF55FF,FFADFF,FFF7FF,FFFFFF,FFFFFF
1920 DATA F1F1F1,02003F,0008FF,0022FF,0055FF
1930 LABEL"M":DATA      0,20, 0,19, 1,18, 1, 2, 0, 1, 0, 0
1940 DATA  6, 0,12,13,19, 0,24, 0,24, 1,23, 2,23,18,24,19
1950 DATA 24,20,16,20,16,19,17,18,17, 9,11,20,10,20, 4, 8
1960 DATA  4,18, 5,19, 5,20, 0,20,-1,-1
1970 LABEL"A":DATA      0,20, 0,18, 1,18,10, 0,14, 0,23,18
1980 DATA 24,18,24,20,15,20,15,18,17,18,15,14, 5,14, 4,18
1990 DATA  5,18, 5,20, 0,20,99,99, 6,12,10, 5,14,12, 7,12
2000 DATA -1,-1
2010 LABEL"N":DATA      0,20, 0,19, 1,18, 1, 2, 0, 1, 0, 0
2020 DATA  3, 0,17,14,17, 2,16, 1,16, 0,24, 0,24, 1,23, 2
2030 DATA 23,18,24,19,24,20,20,20, 7, 7, 7,18, 8,19, 8,20
2040 DATA  0,20,-1,-1
2050 LABEL"U":DATA      0, 0, 0, 1, 1, 2, 1,14, 4,17, 9,20
2060 DATA 14,20,19,18,23,14,23, 9,23, 2,24. 1,24, 0,16, 0
2070 DATA 16, 1,17, 2,17,14,16,16,14,18,10,18, 8,16, 7,14
2080 DATA  7, 2, 8, 1, 8, 0, 0, 0,-1,-1
2090 LABEL"L":DATA      0, 0, 0, 1, 2, 3, 2,17, 0,19, 0,20
2100 DATA 24,20,24,15,22,15,22,16,20,18,10,18, 8,16, 8, 3
2110 DATA 10, 1,10, 0, 0, 0,-1,-1
2120 LABEL"R":DATA      0, 0, 0, 1, 1, 2, 1,19, 0,19, 0,20
2130 DATA  8,20, 8,19, 7,18, 7,12,10,12,16,18,15,19,15,20
2140 DATA 24,20,24,19,22,18,16,12,18,11,22, 9,23, 6,21, 2
2150 DATA 15, 0, 0, 0,99,99, 7, 2, 7,10,14,10,17, 7,17, 5
2160 DATA 14, 2, 7, 2,-1,-1
2170 LABEL"E":DATA      0, 0, 0, 1, 2, 3, 2,17, 0,19, 0,20
2180 DATA 24,20,24,16,23,16,21,18,10,18, 8,16, 8,12, 9,11
2190 DATA 16,11,17,12,18,12,18, 8,17, 8,16, 9, 9, 9, 8, 8
2200 DATA  8, 4,10, 2,21, 2,23, 4,24, 4,24, 0, 0, 0,-1,-1
2210 LABEL" ":DATA     -1,-1
2220 DATA    0, 39,113,175,300
```

リスト2

```
F000 21 10 27 C3 0C F0 21 50 : 88
F008 C3 C3 0C F0 E5 21 00 00 : 88
F010 1E 7A D9 21 3F 01 1E 7A : 6A
F018 CD 81 FA DD 21 80 FC 11 : D3
F020 78 00 ED 4B 36 FC CD 19 : C8
F028 F4 E1 DD 75 05 DD 74 06 : 83
F030 21 C8 32 22 52 FC 21 00 : AC
F038 00 22 50 FC 21 80 32 22 : 63
F040 54 FC 3E 01 32 38 FC CD : C2
F048 59 F3 3E 00 32 38 FC CD : BD
F050 87 F0 FE 1B C8 21 00 00 : 79
F058 CD 2D F8 11 3C FC 01 A0 : DC
F060 32 CD 51 F8 21 B0 31 22 : 6C
F068 52 FC CD E2 F3 01 AB 31 : CD
F070 11 03 FC CD 51 F8 3E 01 : 65
F078 32 56 FC 21 C8 32 22 52 : 13
---------------------------------
SUM: 24 C7 DA 84 94 4F 04 FC 4BBB

F080 FC 06 20 CD F4 F1 C9 CD : 6A
F088 A1 F7 CD 4E F7 3A 3B FC : 1B
F090 47 CD 10 F2 10 FB CD EB : D9
F098 F2 DD 6E 05 DD 66 06 CD : 58
F0A0 2D F8 11 3C FC 01 A0 32 : 41
F0A8 CD 51 F8 CD E2 F3 CD 1D : A2
F0B0 03 FE 1B C8 DD 7E 1F FE : 5C
F0B8 FF C2 87 F0 C9 21 00 00 : 22
F0C0 1E 7A D9 21 3F 01 1E 7A : 6A
F0C8 CD 81 FA DD 21 80 FC 11 : D3
F0D0 78 00 ED 4B 36 FC CD 19 : C8
F0D8 F4 3E 00 32 38 FC 3E 01 : D7
F0E0 DD 77 1D 3E 0F DD 77 07 : 19
F0E8 DD 77 09 CD EF F0 C9 CD : 9F
F0F0 A1 F7 CD 4E F7 3A 3B FC : 1B
F0F8 47 CD 10 F2 DD 7E 1D FE : 8C
---------------------------------
SUM: CB 9B D9 99 FC 1D 20 41 601E

F100 00 CA 40 F1 FE 01 C2 26 : E2
F108 F1 11 13 FC C5 01 26 32 : 2F
F110 CD 51 F8 DD 6E 19 26 00 : A0
F118 0E 00 CD D5 F1 3E 00 DD : BC
F120 77 1D C1 C3 40 F1 11 21 : 7B
F128 FC C5 01 26 32 CD 51 F8 : 30
F130 DD 6E 19 26 00 0E 01 CD : 66
F138 D5 F1 3E 00 DD 77 1D C1 : 36
F140 DD 7E 1E FE 00 CA 84 F1 : B6
F148 FE 01 C2 6A F1 11 1A FC : 43
F150 C5 01 26 32 CD 51 F8 DD : 11
F158 6E 1A 26 00 0E 00 CD D5 : 5E
F160 F1 3E 00 DD 77 1E C1 C3 : 25
F168 84 F1 11 2B FC C5 01 26 : 99
F170 32 CD 51 F8 DD 6E 1A 26 : D3
F178 00 0E 01 CD D5 F1 3E 00 : E0
---------------------------------
SUM: A6 11 C0 15 62 0A 0B 8A 9F98

F180 DD 77 1E C1 DD 7E 08 FE : 94
F188 0F C2 94 F1 3E 02 DD 77 : EA
F190 1D C3 A6 F1 FE 0A C2 A6 : E7
F198 F1 DD 7E 07 FE 08 C2 A6 : C1
F1A0 F1 3E 01 DD 77 1E DD 7E : FD
F1A8 0A FE 0F C2 B6 F1 3E 02 : C0
F1B0 DD 77 1E C3 C8 F1 FE 0A : F6
F1B8 C2 C8 F1 DD 7E 09 FE 08 : E5
F1C0 C2 C8 F1 3E 01 DD 77 1D : 2B
F1C8 05 C2 F9 F0 CD 1D 03 FE : 9B
F1D0 1B C8 C3 EF F0 CD FB F7 : 44
F1D8 CD 1D 03 FE 1B C8 16 00 : E4
F1E0 CD D3 F7 7A E6 01 B9 CA : 7B
F1E8 D8 F1 01 26 32 11 09 FC : 38
F1F0 CD 51 F8 C9 C5 CD A1 F7 : 09
F1F8 CD 4E F7 48 3A 3B FC 47 : 12
---------------------------------
SUM: 82 26 8C B5 7A 44 6A 69 646E

F200 CD 5C F2 10 FB CD EB F2 : D0
F208 CD E2 F3 41 10 E7 C1 C9 : 64
F210 E5 D5 C5 CD B9 F4 CD 63 : 29
F218 F4 DD 7E 1F FE FF CA 57 : 8C
F220 F2 2A 50 FC 23 23 23 22 : F3
F228 50 FC DD 5E 03 CB 2B 16 : 96
F230 00 CB 7B CA 38 F2 16 FF : 4F
F238 DD 6E 05 DD 66 06 B7 ED : 3D
F240 52 D2 51 F2 CB 7A C2 51 : BF
F248 F2 2A 50 FC 3E FF DD 77 : F9
F250 1F DD 75 05 DD 74 06 C1 : 8E
F258 D1 E1 97 C9 C5 CD 6D F2 : 03
F260 CD 63 F4 2A 50 FC 23 23 : E0
F268 22 50 FC C1 C9 E5 D5 C5 : 77
F270 DD E5 E1 01 07 00 09 06 : BA
F278 02 DD 7E 0B FE 80 D2 86 : 3E
---------------------------------
SUM: 94 7E D1 F1 4F A8 43 88 D0F6

F280 F2 3E 81 DD 77 0B 0E 00 : 1E
F288 16 00 CA 8F F2 0E 0F 7E : FC
F290 B9 DA 9B F2 C2 9F F2 14 : 87
F298 C3 A0 F2 3C C3 A0 F2 3D : 23
F2A0 77 23 10 EB 3E 02 BA C2 : 51
F2A8 AF F2 3E 80 DD 77 0B 06 : C4
F2B0 02 DD 7E 0C FE 80 D2 BE : 77
F2B8 F2 3E 81 DD 77 0C 0E 00 : 1F
F2C0 16 00 CA C7 F2 0E 0F 7E : 34
F2C8 B9 DA D3 F2 C2 D7 F2 14 : F7
F2D0 C3 D8 F2 3C C3 D8 F2 3D : 93
F2D8 77 23 10 EB 3E 02 BA C2 : 51
F2E0 E7 F2 3E 80 DD 77 0C C1 : B8
F2E8 D1 E1 C9 E5 D5 C5 3A 56 : 8A
F2F0 FC FE 00 CA 55 F3 FE 01 : 0B
F2F8 C2 1A F3 16 04 06 1C ED : F8
---------------------------------
SUM: 1D A8 BE 13 3E 51 B3 EB F1EA

F300 51 05 3E 00 ED 79 04 14 : 12
F308 ED 51 05 3E 03 ED 79 04 : EE
F310 14 ED 79 05 3E 1E ED 79 : 41
F318 3E 01 3C FE 12 C2 32 F3 : 72
F320 3E 00 32 56 FC ED 4B 54 : 4E
F328 FC 11 09 FC CD 51 F8 C3 : EB
F330 55 F3 32 56 FC CB 27 CB : 89
F338 27 CB 27 C6 32 16 04 06 : 31
F340 1C ED 51 05 ED 79 04 16 : DF
F348 0A ED 51 05 3A 56 FC 57 : 30
F350 3E 10 92 ED 79 C1 D1 E1 : B9
F358 C9 E5 D5 C5 ED 4B 54 FC : D0
F360 3E 01 32 56 FC 11 F7 FB : C6
F368 CD 51 F8 06 20 CD A1 F7 : A1
F370 CD 4E F7 48 3A 3B FC 47 : 12
F378 CD B9 F4 CD 63 F4 10 F8 : A6
---------------------------------
SUM: 18 3B AA DC 7D 4D D3 E7 466B

F380 41 10 EA 3E 01 32 56 FC : FE
F388 ED 4B 54 FC 11 FD FB CD : 5E
F390 51 F8 C1 D1 E1 C9 E5 D5 : 3F
F398 C5 ED 4B 54 FC 3E 01 32 : BE
F3A0 56 FC 11 F7 FB CD 51 F8 : 6B
```

▶うーん，EXIV（トヨタの）はいい。結構速いしね。1800だけどフルグレードだから満足だ。でもさっそくこすっちまった……。
諸岡　明(21)神奈川県

```
F3A8 06 20 48 26 02 DD 21 80 : 14
F3B0 FC CD A1 F7 CD 4E F7 3A : AD
F3B8 3B FC 47 CD B9 F4 CD 63 : 28
F3C0 F4 10 F8 11 20 00 DD 19 : 23
F3C8 25 C2 B1 F3 41 10 DB 3E : F5
F3D0 01 32 56 FC ED 4B 54 FC : 0D
F3D8 11 FD FB CD 51 F8 C1 D1 : B1
F3E0 E1 C9 E5 D5 C5 2A 50 FC : 9F
F3E8 CD 2D F8 21 3D FC 11 3C : 99
F3F0 FC ED A0 ED A0 ED A0 1B : BE
F3F8 1B 06 03 1A FE 20 C2 03 : 21
---------------------------------
SUM: C7 0F 05 0A B1 A8 FD 5F D350

F400 F4 3E 30 12 13 10 F4 1B : A6
F408 1B 3E 2E 12 1B 1B ED 4B : 07
F410 52 FC CD 51 F8 C1 D1 E1 : D7
F418 C9 E5 D5 C5 DD E5 E1 06 : F1
F420 20 3E 00 77 23 10 FC 26 : 2A
F428 64 DD 74 00 DD 74 17 DD : FA
F430 77 01 DD 77 18 DD 77 13 : 4B
F438 DD 77 12 26 78 DD 74 13 : 68
F440 DD 77 14 21 F0 00 DD 75 : CB
F448 15 DD 74 16 DD 72 0D DD : B5
F450 73 0E 7B 1E 3E 93 DD 77 : 3F
F458 02 C1 DD 70 19 DD 71 1A : 91
F460 D1 E1 C9 E5 D5 C5 21 A0 : BB
F468 FC DD 7E 0D 5F 87 87 87 : 58
F470 85 6F 22 4A FC DD 56 07 : 96
F478 DD 5E 08 CD 12 F9 2A 4A : 8F
---------------------------------
SUM: 98 9E B4 1C F9 13 F1 D1 4F11

F480 FC 3E 04 85 6F 22 4A FC : 9A
F488 DD 56 09 DD 5E 0A CD 12 : 60
F490 F9 CD B6 F6 CD 91 F5 3A : FF
F498 38 FC FE 00 CA A4 F4 3E : D2
F4A0 00 DD 77 03 CD E1 F6 11 : 0C
F4A8 B8 FC 21 A0 FC 01 18 00 : 8A
F4B0 ED B0 11 A0 FC C1 D1 E1 : BD
F4B8 C9 E5 D5 C5 3E 00 DD 77 : DA
F4C0 10 DD 56 0D DD 66 19 DD : 89
F4C8 6E 1A CD D3 F7 DD 5E 0B : 65
F4D0 DD 46 07 DD 4E 08 60 69 : 26
F4D8 CB 4A C2 E3 F4 CD 6B F5 : DB
F4E0 C3 F0 F4 DD 7E 0F CB 4F : 2B
F4E8 C2 ED F4 1E 00 CD 42 F5 : C5
F4F0 DD 73 0B DD 70 07 DD 71 : FD
F4F8 08 B7 ED 42 CA 04 F5 3E : EF
---------------------------------
SUM: 08 59 0B 1A 35 03 DD 28 090B

F500 01 DD 77 10 DD 5E 0C DD : 89
F508 46 09 DD 4E 0A 60 69 CB : 18
F510 42 C2 1A F5 CD 6B F5 C3 : 03
F518 27 F5 DD 7E 0F CB 47 C2 : 5A
F520 24 F5 1E 00 CD 42 F5 DD : 18
F528 73 0C DD 70 09 DD 71 0A : 2D
F530 DD 72 0F B7 ED 42 CA 3E : 4C
F538 F5 3E 01 DD 77 10 C1 D1 : 2A
F540 E1 C9 3E 7F A3 5F CB 73 : A7
F548 C2 4C F5 1C 3E 0F B9 CA : EF
F550 6A F5 0C 3E 0D B9 F2 63 : C4
F558 F5 3E 00 B8 CA 6A F5 05 : 19
F560 C3 6A F5 3E 0F B8 CA 6A : 5B
F568 F5 04 C9 3E 00 BB C8 1D : A0
F570 3E 00 B9 CA 77 F5 0D 3E : 78
F578 04 B9 FA 89 F5 04 04 CB : 08
---------------------------------
SUM: 15 BD 06 35 30 62 B0 58 3EA3

F580 60 CA 90 F5 06 0F C3 90 : 17
F588 F5 3E 00 B8 CA 90 F5 05 : 3F
F590 C9 E5 D5 C5 11 A0 FC DD : D2
F598 7E 0D 87 87 87 83 5F 06 : 08
F5A0 04 0E 00 13 DD 7E 0F E6 : 75
F5A8 01 DD 6E 0F CB 2D AD 0F : 0F
F5B0 32 39 FC 1A CB 7F C2 C9 : 56
F5B8 F5 B9 DA C9 F5 4F 60 C2 : B7
F5C0 C9 F5 3A 39 FC 3C 32 39 : D4
F5C8 FC 13 13 10 E6 CB 24 7B : 82
F5D0 94 5F DD 7E 02 81 DD 4E : FC
F5D8 0E 47 DD 34 04 DD 34 04 : 7F
F5E0 DD 34 04 B9 DA 86 F6 91 : B5
F5E8 6F DD 7E 02 95 3C DD 77 : F1
F5F0 02 DD 6E 03 CB 7D CA 03 : 65
F5F8 F6 2C 2C FA 0C F6 2E 00 : 78
---------------------------------
SUM: 73 9F 53 B1 FE D5 23 09 7321

F600 C3 0C F6 2D 2D CB 7D CA : 31
F608 0C F6 2E 00 DD 75 03 DD : 62
F610 7E 10 FE 00 C2 1D F6 DD : 3E
F618 77 04 C3 43 F6 78 91 6F : EF
F620 4F CD 04 F8 79 CB 2D 87 : 10
F628 87 6F 3A 39 FC CB 7F C2 : 71
F630 3A F6 FE 00 CA 3A F6 7D : A5
F638 87 6F 7D EE FF DD 86 04 : C7
F640 DD 77 04 CB 4C CA 4A F6 : 79
F648 1B 1B 3E 00 32 3A FC 1A : F6
F650 6F 13 13 1A 95 F2 5A F6 : 86
F658 EE FF B9 D2 63 F6 3E 01 : 10
F660 32 3A FC 1B 1A 6F 3E 18 : 62
F668 83 5F 1A 67 7D 94 6F 6F : 52
F670 3A 3A FC FE 00 7D CA 7A : 2F
F678 F6 87 EE FF 3C DD 86 03 : 0C
---------------------------------
SUM: 95 B5 AC C5 49 CB 0A C8 056F

F680 DD 77 03 C3 B2 F6 CD FB : 8A
F688 F7 0E 00 DD 7E 03 CB 7F : AD
F690 CA 97 F6 EE FF 0E 01 CB : 1E
F698 2F CB 2F CB 2F CB 2F 6F : 8C
F6A0 CB 2D CB 41 CA A9 F6 EE : 5B
F6A8 FF 95 6F DD 7E 03 95 DD : D3
F6B0 77 03 C1 D1 E1 C9 DD 7E : 11
F6B8 04 CB 2F CB 2F CB 2F DD : CF
F6C0 86 02 DD 77 02 DD 6E 17 : 40
F6C8 DD 66 18 DD 4E 03 CB 29 : 7D
F6D0 06 00 CB 79 CA D9 F6 06 : E9
F6D8 FF 09 DD 75 00 DD 74 01 : AC
F6E0 C9 E5 D5 C5 06 00 DD 4E : 79
F6E8 03 CB 79 CA F0 F6 06 FF : FC
F6F0 CB 29 CB 29 CB 29 DD E5 : 9E
F6F8 E1 11 11 00 19 E5 DD 7E : 5C
---------------------------------
SUM: F2 D2 19 0D AA AC 9F D1 68CC

F700 0E 5F 1C D9 C1 D6 05 5F : 5D
F708 16 03 0A 6F 03 0A 67 E5 : EB
F710 D9 E1 3E 01 BC C2 1E F7 : 8C
F718 3E 3F BD DA 21 F7 CD 8C : 85
F720 FA B7 ED 42 F2 2D F7 C5 : BB
F728 01 68 01 09 C1 E5 D9 E1 : D3
F730 0B 7D 02 03 7C 02 03 3E : 4C
F738 01 BC C2 43 F7 3E 3F BD : F3
F740 DA 46 F7 CD 81 FA 15 C2 : 36
F748 0A F7 C1 D1 E1 C9 E5 D5 : F7
F750 C5 DD 7E 0D 87 87 57 87 : 19
F758 82 57 87 82 87 82 26 FD : 0E
F760 6F EB DD E5 E1 01 03 00 : 01
F768 ED B0 EB 22 4A FC DD 56 : 23
F770 07 DD 5E 08 CD 90 F8 DD : 7C
F778 6E 00 DD 66 01 DD 5E 02 : EF
---------------------------------
SUM: 3E C3 93 56 30 21 16 B8 7504

F780 CD 70 F9 3A 4A FC C6 28 : A4
F788 32 4A FC DD 56 09 DD 5E : EF
F790 0A CD 90 F8 2B 2B 2B DD : BD
F798 5E 02 CD 70 F9 C1 D1 E1 : 09
F7A0 C9 E5 D5 DD 7E 0D 87 87 : F9
F7A8 57 87 82 57 87 82 87 82 : C9
F7B0 26 FD 6F 5E 23 56 23 7E : 0A
F7B8 23 EB ED 53 4A FC 5F CD : C0
F7C0 7D F9 3A 4A FC C6 28 32 : 16
F7C8 4A FC 2B 2B 2B CD 7D F9 : 0A
F7D0 D1 E1 C9 C5 E5 3E 02 BA : 1F
F7D8 CA F8 F7 01 00 1C 3E 0E : 22
F7E0 82 ED 79 05 ED 58 16 00 : 48
F7E8 7C A3 C2 EF F7 16 02 7D : 5C
F7F0 A3 C2 F5 F7 14 C3 F8 F7 : 17
F7F8 E1 C1 C9 E5 D5 C5 21 F4 : FF
---------------------------------
SUM: B4 BE 23 6F 0F B5 45 F3 C2E8

F800 FB C3 16 F8 E5 D5 C5 4F : 9A
F808 81 FE 0F DA 10 F8 3E 0F : BD
F810 32 F3 FB 21 F1 FB 06 1C : 4F
F818 1E 06 16 03 7E 23 ED 59 : 24
F820 05 ED 79 04 1C 15 C2 1C : 7E
F828 F8 C1 D1 E1 C9 E5 D5 C5 : B3
F830 11 40 FC 06 05 CD 65 F8 : 82
F838 C6 30 12 1B 10 F7 13 06 : 43
F840 04 1A FE 30 C2 4D F8 3E : 91
F848 20 12 13 10 F4 C1 D1 E1 : BC
F850 C9 E5 D5 C5 1A FE 00 CA : 2A
F858 61 F8 ED 79 13 03 C3 54 : EC
F860 F8 C1 D1 E1 C9 D5 C5 3E : 0C
F868 0D 01 00 00 11 00 A0 B7 : 76
F870 ED 52 D2 79 F8 19 C3 7A : D8
F878 F8 0C CB 21 CB 10 CB 3A : D0
---------------------------------
SUM: D8 01 CF F5 DE B6 84 98 5621

F880 CB 1B 3D C2 6F F8 7D CB : 94
F888 38 CB 19 60 69 C1 D1 C9 : 40
F890 E5 D5 C5 D5 21 59 FC ED : B7
F898 5B 4A FC 01 28 00 ED B0 : 67
F8A0 D1 D5 ED 4B 4A FC 3E 06 : 68
F8A8 81 4F 1E 09 0A D6 04 67 : 42
F8B0 03 0A D6 1D 6F 0B CD CE : 15
F8B8 F9 3E 04 84 02 03 3E 1D : 1F
F8C0 85 02 03 1D C2 AC F8 CB : D8
F8C8 2A ED 4B 4A FC 3E 04 81 : 6B
F8D0 4F 0A D6 04 67 03 0A 0B : B2
F8D8 D6 1D 6F CD CE F9 7C C6 : 38
F8E0 04 02 03 3E 1D 85 02 D1 : BC
F8E8 53 ED 4B 4A FC 1E 12 0A : 0B
F8F0 D6 07 67 03 0A D6 05 0B : 37
F8F8 EE FF 6F CD CE F9 7C C6 : 32
---------------------------------
SUM: 80 7C B3 7D CA 4A 9B 52 2A59

F900 07 02 03 7D EE FF C6 05 : 41
F908 02 03 1D C2 EF F8 C1 D1 : 5D
F910 E1 C9 E5 D5 C5 D5 ED 5B : 46
F918 4A FC 21 67 FC 01 04 00 : CF
F920 ED B0 D1 D5 ED 4B 4A FC : C1
F928 1E 02 0A D6 04 67 03 0A : 78
F930 D6 1D 6F 0B CD CE F9 3E : 3F
F938 04 84 02 03 3E 1D 85 02 : 6F
F940 03 1D C2 2A F9 D1 53 ED : 16
F948 4B 4A FC 1E 02 0A D6 07 : 98
F950 67 03 0A D6 05 0B EE FF : 47
F958 6F CD CE F9 7C C6 07 02 : 4E
F960 03 7D EE FF C6 05 02 03 : 3D
F968 1D C2 4D F9 C1 D1 E1 C9 : 61
F970 E5 D5 C5 01 81 FA ED 43 : 2B
F978 BE F9 C3 87 F9 E5 D5 C5 : 79
---------------------------------
SUM: 00 61 CB CB 17 CB 06 40 B13E

F980 01 8C FA ED 43 BE F9 ED : 5B
F988 4B 4A FC 16 11 22 46 FC : 1C
F990 ED 53 48 FC 0A 16 00 5F : 03
F998 19 ED 5B 48 FC 03 0A 03 : B5
F9A0 83 5F C5 D9 C1 2A 46 FC : AD
F9A8 0A 03 16 00 CB 7F CA B3 : EA
F9B0 F9 16 FF 5F 19 ED 5B 48 : 16
F9B8 FC 0A 03 83 5F CD 81 FA : 33
F9C0 3A 49 FC 3D 32 49 FC C2 : F5
F9C8 A2 F9 C1 D1 E1 C9 D5 C5 : 71
F9D0 3E 00 BA CA 0E FA 3E 0F : 17
F9D8 BA C2 EB F9 7D 6C EE FF : 36
F9E0 FE FF C2 E7 F9 3E 00 67 : 44
F9E8 C3 0E FA 4C 3E 0E 92 47 : 3C
F9F0 CD 11 FA 59 4D 42 CD 11 : 9E
F9F8 FA 7B 91 5F 4C 42 CD 11 : D1
---------------------------------
SUM: 30 35 1F BE CC A4 5E A1 B072

FA00 FA 61 4D 3E 0E 92 47 CD : 9A
FA08 11 FA 7C 81 6F 63 C1 D1 : 6C
FA10 C9 E5 05 3E 00 32 4F FC : 6E
FA18 CB 79 CA 26 FA 79 EE FF : 94
FA20 4F 3E 01 32 4F FC 21 E3 : 0F
FA28 FB 7D 80 6F 7E 21 00 00 : 06
FA30 06 00 0F 30 01 09 CB 21 : 3B
FA38 CB 10 0F 30 01 09 CB 21 : 10
FA40 CB 10 0F 30 01 09 CB 21 : 10
FA48 CB 10 0F 30 01 09 CB 21 : 10
FA50 CB 10 0F 30 01 09 CB 21 : 10
FA58 CB 10 0F 30 01 09 CB 21 : 10
FA60 CB 10 0F 30 01 09 CB 21 : 10
FA68 CB 10 0F 30 01 09 CB 21 : 10
FA70 CB 10 3A 4F FC FE 01 C2 : 21
FA78 7E FA 7C EE FF 67 4C E1 : 75
---------------------------------
SUM: C5 EE 47 81 47 6A 0B 27 F2A8

FA80 C9 3E C3 32 80 FB 32 64 : 0D
FA88 FB C3 94 FA 3E 83 32 80 : BF
FA90 FB 32 64 FB C5 D5 E5 D9 : E4
FA98 C5 D5 E5 D9 7B D9 93 DA : 19
FAA0 A3 FA D9 7B FE C8 D2 6D : F6
FAA8 FB 3C 32 4E FC 3E 08 32 : 2B
FAB0 4D FC D9 08 7B 08 E5 D9 : 6B
FAB8 C1 E5 B7 ED 42 F2 CC FA : 44
FAC0 60 69 C1 B7 ED 42 C5 3E : 73
FAC8 78 32 4D FC C1 7B D9 93 : 9B
FAD0 D9 3C 06 00 4F 54 5D 69 : 84
FAD8 B7 CB 1D 26 00 D9 7D E6 : 01
FAE0 07 32 4C FC CB 3C CB 1D : 70
FAE8 CB 3C CB 1D CB 3C CB 1D : DE
FAF0 E5 26 00 7B E6 F8 6F 7B : 4E
FAF8 E6 07 54 5D 29 29 19 E5 : EE
---------------------------------
SUM: 35 5C D7 88 57 AF FD C3 255C

FB00 67 2E 00 B4 CB 14 CB 14 : 07
FB08 CB 14 D1 19 D1 19 3A 35 : 22
FB10 FC B4 67 44 4D 16 03 3A : FB
FB18 4D FC FE 78 C2 21 FB 16 : B3
FB20 0B 7A 32 95 FB 3A 4C FC : C9
FB28 B7 17 17 17 57 C3 5E FB : 6F
FB30 D9 19 CD 75 FB 09 D9 ED : FE
FB38 59 08 67 24 08 3A 4E FC : 78
FB40 BC CA 6D FB 08 7C 08 7C : F6
FB48 E6 07 C2 58 FB 60 69 01 : CC
FB50 D8 37 B7 ED 42 C3 5C FB : 0F
FB58 21 00 08 09 44 4D ED 58 : 08
FB60 3E 38 AA F6 C3 32 69 FB : 6F
FB68 CB C3 C3 30 FB E1 D1 C1 : EF
FB70 D9 E1 D1 C1 C9 D9 D9 B7 : 7E
FB78 ED 42 F8 D9 3E 38 AA F6 : 16
---------------------------------
SUM: D9 CA D7 D7 4E B4 4B B2 7CBB

FB80 C3 32 85 FB CB C3 3A 4D : 8A
FB88 FC 82 6F E6 38 57 CB 75 : A2
FB90 CA 76 FB ED 59 03 ED 58 : C9
FB98 C3 76 FB 0F 04 0D 1A 0D : 7B
FBA0 20 0C 24 09 30 08 38 13 : DC
FBA8 3C 13 3F 00 3E 01 37 00 : 04
FBB0 29 02 25 04 1D 01 14 00 : 86
FBB8 05 03 01 0A 00 0F 04 0D : 33
FBC0 03 0C 1A 0C 20 0B 24 08 : 8C
FBC8 30 07 32 0D 38 0D 3E 01 : FA
FBD0 3D 02 35 02 28 02 24 03 : C7
FBD8 14 02 0C 00 08 03 02 09 : 38
FBE0 00 0D 03 1B 35 4F 68 80 : 97
FBE8 96 AB BE CE DD E9 F3 F9 : 7F
FBF0 FE 1E 13 0F 1E 13 00 52 : C1
FBF8 45 41 44 59 00 47 4F 20 : D9
---------------------------------
SUM: 33 F2 18 60 A3 F2 C5 47 9ED0

FC00 21 20 00 4C 41 50 20 3A : 78
FC08 00 20 20 20 20 20 20 20 : E0
FC10 20 20 00 50 55 53 48 20 : A0
FC18 41 00 50 55 53 48 20 42 : E3
FC20 00 52 45 4C 45 41 53 45 : 01
FC28 20 41 00 52 45 4C 45 41 : CA
FC30 53 45 20 42 00 C0 40 20 : 1A
FC38 00 00 00 00 00 00 00 00 : 00
FC40 00 00 00 00 00 00 00 00 : 00
FC48 00 00 00 00 00 00 00 00 : 00
FC50 00 00 00 00 00 00 00 00 : 00
FC58 00 00 00 00 00 00 00 00 : 00
FC60 00 00 00 00 00 00 00 00 : 00
FC68 00 00 00 00 00 00 00 00 : 00
FC70 00 00 00 00 00 00 00 00 : 00
FC78 00 00 00 00 00 00 FF 00 : FF
---------------------------------
SUM: F5 38 D5 F1 93 58 7F 62 EDC9
```

▶最近になってメモリは4～6Mバイト必要だな。と思い財布と相談したが話にならなかった……。
関口　敬文(16)兵庫県

BACK ISSUES バックナンバー案内

ここには1990年9月号から1991年8月号までをご紹介しました。現在1990年10, 11, 1991年1～8月号の在庫がございます。バックナンバーおよび定期購読の申し込み方法については，178ページを参照してください。

1990

9月号（品切れ）
特集1　日本語を処理するための序章
特集2　ADVANCED 2D GRAPHICS
連載　ショートプロぱーてぃ/Z80's Bar/DōGA・CGA
X-BASIC調理実習/マシン語プログラミング
PurePASCAL/ハードウェア工作入門
●清水和人流プログラミング道場
LIVE in '90　風の谷のナウシカ/ラジオ体操第一
THE SOFTOUCH　T&T/D-Again/シムシティー/ギャラガ'88ほか
全機種共通システム　BILLIARDS

10月号
特集　電子音楽術入門
連載　ショートプロぱーてぃ/Z80's Bar/DōGA・CGA
マシン語プログラミング/ハードウェア工作入門
清水和人流プログラミング道場
●荻窪圭の大人のためのX68000
●中森章のようこそここへC言語

LIVE in '90　Rise And Fall/PARADOX/キューピー3分クッキング
THE SOFTOUCH　ワールドコート/ルーンワース/闇の血族/提督の決断
全機種共通システム　ライブラリアンWLB

11月号
特集　理科系のGAME REVIEW
連載　Z80's Bar/DōGA・CGA/カードゲーム
マシン語プログラミング/ハードウェア工作入門
PurePASCAL/X-BASIC調理実習
ようこそここへC言語/INTEGRAL XI
●荻窪圭の大人のためのX68000
LIVE in '90　ピラミッドソーサリアン/ザ・スキーム
THE SOFTOUCH SPECIAL　ラグーン/幻獣鬼/サイバリオン/GUNSHIP他
全機種共通システム　スクリーンエディタEDC-T

12月号（品切れ）
特集　XCのための傾向対策
連載　X-BASICプログラミング調理実習/ハードウェア工作入門
マシン語プログラミング/ショートプロぱーてぃ/Z80's Bar
大人のためのX68000/ようこそここへC言語/INTEGRAL XI
●シミュレーションプログラミング入門
●特別企画アナログジョイステックの制作
LIVE in '90　グラディウスIII/メタルサイト
THE SOFTOUCH SPECIAL　イメージファイト/ジェミニウイング/NAIOUS他
全機種共通システム　STACKコンパイラ

1991

1月号
特集　急接近! SX-WINDOW
特別付録　謹賀新年PRO-68K（5"2HD）
連載　ハードウェア工作入門/シミュレーションプログラミング入門
DōGA・CGA/ショートプロぱーてぃ/大人のためのX68000
PurePASCAL/清水和人流プログラミング道場/X-BASIC調理実習
LIVE in '91　めぞん一刻/涙で綴るパパへの手紙
THE SOFTOUCH　ソル・フィース/銀英伝II/続ダンジョン・マスター他
製作紹介　光磁気ディスクCZ-6MOI
全機種共通システム　ブロックアクションゲームCOLUMNS

2月号
特集1　グラフィックの"実験的"手法
特集2　SX!WINDOWプログラミング
連載　ハードウェア工作入門/シミュレーションプログラミング入門
マシン語プログラミング/大人のためのX68000/Z80's Bar
ショートプロぱーてぃ/INTEGRAL XI/ようこそここへC言語
●1990年度GAME OF THE YEARノミネート発表
LIVE in '91　Misty Blue/スプーンおばさん
THE SOFTOUCH　栄冠は君に/KLAX/ダイナマイト・デューク他
全機種共通システム　ダイスゲームKISMET

3月号
特集　MIDI & MUSIC PROCESSING
連載　ハードウェア工作入門/シミュレーションプログラミング入門
マシン語プログラミング/大人のためのX68000/Z80's Bar
ショートプロぱーてぃ/DōGA・CGA/C言語/PurePASCAL
●SXLIFE完結編/ウィンドウシステム大比較
●周辺機器新製品紹介
LIVE in '91　戦いの兜/LITTLE WING/リゾ・ラバ/花
THE SOFTOUCH　アトミック・ロボキッド/スペースラローグ他
全機種共通システム　アクションゲームMUD BALLIN'

4月号
特集　人とゲームのインタフェイス
連載　DōGA・CGA/シミュレーションプログラミング入門
ハードウェア工作入門/ようこそここへC言語/Z80's Bar
ショートプロぱーてぃ/清水和人流プログラミング道場
●新連載　吾輩はX68000である/よいこのSX-WINDOW講座
●決定!　1990年度GAME OF THE YEAR
LIVE in '91　Easy Come, Easy Go!/シシリエンヌ
THE SOFTOUCH　メルヘンメイズ/中華大仙/スライス他
全機種共通システム　SLANG用カードゲームDOBON

5月号
特集　新登場!　X68000XVI/XVI-HD
特別付録　黄金週間PRO-68K（5"2HD）
第6回　言わせてくれなくちゃだワ
連載　ハードウェア工作/ようこそここへC言語
大人のためのX68000/X68000マシン語プログラミング
ショートプロぱーてぃ/マシン語カクテル in Z80's Bar
LIVE in '91　ブービーキッズ/NO.NEW YORK
THE SOFTOUCH　マーブル・マッドネス/シグナトリー/石道他
全機種共通システム　実数型コンパイラ言語REAL

6月号
特集　初心者のための環境構成術
創刊9周年記念Oh!Xアンケート結果大分析大会その1
連載　ハード工作/大人のためのX68000/Z80's Bar/DOGA
ようこそC言語/ショートプロぱーてぃ/SX-WINDOW
吾輩はX68000である/マシン語プログラミング
●響子　in CGわーるど
LIVE in '91　暴れん坊将軍/ナディア/POWER HALL他
THE SOFTOUCH　パロディウスだ!/遙かなるオーガスタ/ノスタルジア他
全機種共通システム　S-OS 6周年記念　Small-C　処理系の移植

7月号
特集　Personal Tool,BASIC
別冊付録　X-BASIC　ポケットリファレンスブック
連載　大人のためのX68000/ハード工作/響子 in CGわーるど
ショートプロぱーてぃ/SX-WINDOW/吾輩はX68000である
ようこそC言語/Z80's Bar/マシン語プログラミング
●XI用ゲーム　The Master of Payment
LIVE in '91　今すぐKISS ME/歩いていこう
THE SOFTOUCH　パロディウスだ!/ファランクス/スコルピウス/AIII他
全機種共通システム　実数型コンパイラ言語REAL ソースリスト編

8月号
特集　印刷の世界へ
連載　大人のためのX68000/SX-WINDOW/ようこそC言語
響子 in CGわ～るど/ハード工作/ショートプロぱーてぃ
吾輩はX68000である/マシン語プログラミング
●X68000カードゲーム　七並べ
●XI用ゲーム　DEFEAT2
LIVE in '91　パワードリフト/イースIII/TURBO OUTRUN
THE SOFTOUCH　黄金の羅針盤/サイレントメビウス/パロディウスだ!他
全機種共通システム　Small-C ライブラリの移植

★(で)のショートプロぱーてぃ その24

小さく小さく，奥へ奥へ

Komura Satoshi 古村 聡

今回もいつもどおりにショートプロぱーてぃのお題は2本。X68000用縮小印字ツール「SPRN」と，X1用スクロールゲーム「MERVEL」です。一方ハンズのほうはリストがなくて，思考ルーチンの基本的考え方の説明に終始しています。

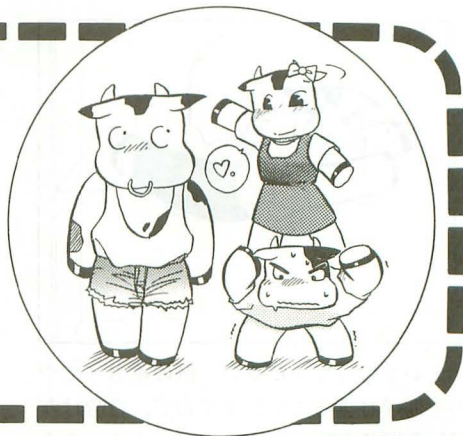



illustration : T.Takahashi

今日も元気だ，サイフが軽い。はっはっはっ。いやあ，ついに私も買ってしまいました。プリンタをね。

いままでプリンタを持ってなかったのかって。実はそうだったんです。編集室にはVPなんとかとか，CZ-8PCなんちゃらとかいうプリンタがたくさんあるので，原稿の打ち出しとかはほとんど編集室でやってたんです。が，それだとやっぱり不便だし，打ち出して原稿をチェックしないと編集さんに迷惑かけるしなあ（……誤字脱字が多いのをプリンタがないせいにしちゃいかんけど）などと思って，ついにBJ-10vなるプリンタを秋葉原で買ってきてしまったのです。ふふんふふん。

で，秋葉原を歩いていたら，さらに“メガドライブ大安売り”と“マーベルランド”などというものがあって，思わずこいつも衝動買いしてしまったのです。だって，マーベルランドはやりたいやりたいと思ってたんですよ。ゲームセンターから消えてしまったあの日から，ずっとずっとやりたかったんだよーっ！ うるうるっ。

おかげさまでゲームで遊んでばかりいるから仕事ははかどらない，お金はない……。しばらくのあいだは究極貧乏になりそうです。いいもん。マーベルランドさえあれば世の中楽しいやい。

文字小さい，うれしい

ではでは，今月の1本目いきます。X68000用縮小印字ツール，SPRN.Cです。

SPRN.C for X68000

（要Cコンパイラ）
大阪府 野崎哲也

シャープ純正の24ピン以上の漢字プリンタ（CZ-8PC3/4/5，CZ-8PK8など）を使って縮小文字印刷をするプログラムです。

このプログラムはCで書かれていますので実行するためにはC compiler PRO-68Kなどが必要になります（ver.1/2のどちらでもOK）。また，GCC（ver.1.36～9）でコンパイルできることも確認しています（コンパイルオプションは自分で設定してね）。

リストはエディタ(ed.xなど)を使って打ち込んでください。その際に行番号はつけないように注意してくださいね。

で，セーブしたら，次はコンパイル。コマンドライン上で，

A> CC /Y SPRN.C

とすると，エラーがなければ，SPRN.Xというファイルができます。これでOK。

コンパイラによっては大量にウォーニング(WARNING)が出てしまうみたいですが，気にしなくても大丈夫です。ちゃんと動きます。

ここまでできたらあとは簡単。普通に，

A> SPRN ファイル名

と入れてやると，プリンタにファイルが縮小文字で印字されて出てきます。

このとき，一緒に/Pと打つと画面上にも文字が表示され，/Bとやると24ドットフォントで印字されるようになっています（ふだんは16ドットフォント)。24ドットでも文字間隔を詰めるので，CZ-8PC3を使っている場合は普通に打つよりも紙の節約になりますね。

……ふうむ。16ドットだとずいぶん小さく打ち出すんですね。16，24のどちらがいいかは個人の好みの問題でしょう。リスト打ち出すのとかに便利そうですね，16ドットのほうは。

このプログラムでは漢字を印字するのにX68000本体のROMのパターンを使っているんですね。てことは16ドットフォントで打つと，基本的には画面に表示される字体と同じものがでてくるわけですね。うーん，本体のフォントってなかなかきれいだなあ，こうやって見ると。

個人的な趣味かもしれないんですけど，X68000の画面のフォントって整ってる感じがしてきれいなんですよね。そういえば，編集室にも「PC-9801の“ー（音引き）”はフニャッとしてて嫌いだあ」とか叫んでいる人がいたなあ。私は98NOTEとかのフォントも斜め線が見やすくて（なぜか，PC-9801のなかでもnシリーズだけは斜め線が微妙に太くなってるんです。不思議だ）好きなんですけどね。むむむ。フォントって本当に大事(ダジャレじゃないけど)。

SPRN印字例

私は（で）です。
これからもよろしく。

私は（で）です。
これからもよろしく。

さて，このプログラムはシャープ純正プリンタのみ対応のプログラムなのですが，リストを見ていると，エスケープシーケンスのところ，つまり，LPTOUT(27)とかの“27”という数字の部分を変えれば，純正以外のプリンタ，たとえばESC/Pプリンタ（私のBJ-10vなんかはこれね）とかPC-PR201系なんかでも動きそうな感じがします。その部分にはしっかり注釈もふってあるみたいなので，純正以外のプリンタの人も移植に挑戦してみてはどうでしょうか。

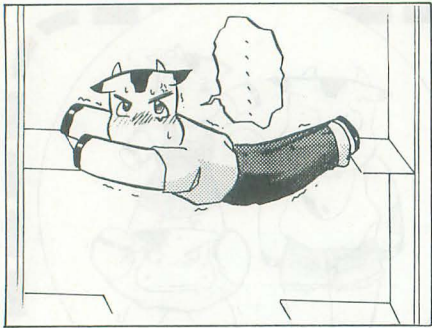
走れゲームの道

2本目のプログラムはX1用のゲーム「MERVEL」です。

MERVEL.BAS for X1

（CZ-8FB01）
埼玉県 根本篤

X1BASIC用のゲームですので，X1全機種で遊べます。グラフィックも使っていないのでNEW-BASICでもOKです。

で，遊び方。自分は@マーク。んで，テンキーの4，5，6を使って障害物（*）を避けながら，@をがんがん地底の奥深く

へといざなってください。例の重力ってやつで勝手に落ちていきますので，自分は左右に動かすだけ。「ニュートンのバカやろう……おうおうおう」，とでも叫びながら地獄の底へ落ちていってください。

方向は4で左，6で右。ただし，一度キーを押すとずっと左へ，右なら右に行きっぱなしなので注意してね。5のキーを押せばまっすぐ下に落ちていきます。奥に行けば行くほど障害物の間隔も，隙間の幅も狭くなっていくんですよね。つらいっ！

ふむふむ。ひさしぶりのスクロールタイプのゲームですね。結構，スクロールゲームは作りやすいと思うんですが，なぜか投稿では最近このテのゲームってあまり多くないんですね。不思議。わりとありがちだから，とか思っているのかな？　いまならお得かもしれないですよん。投稿者の皆さん。あと，最近X1のショートプロ向けの投稿が少なくなってきているので，これまたチャンスかもしれませんよ。

慣性がかかるっていうのは，わりとショートでも多くあるパターンだけど，キー入力が入ったら入りっぱなしというのはめずらしいですね。なんか昔のFM-7とかを思い出してしまうなあ。なんか障害物にぶつかったときに妙な味があるし……，まったく不思議な感じのゲームであります。

え，なになに？　投稿原稿によれば，

「本当は慣性をつけようと思っていたんだけど，テキストで慣性をつけると大きく動きすぎるので，キーが入りっぱなしになるようにしてしまいました。ちっ，妥協しちまったぜ」

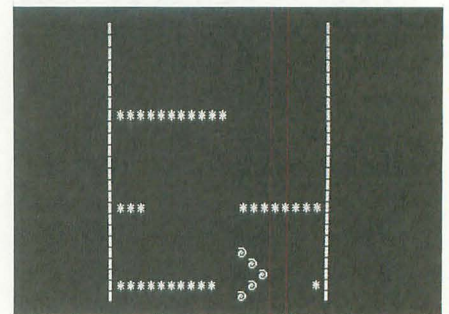
なのだそうなのですね。ん？

「本当はマーベルランドの落ちていく面が作りたかったんだけど。ああ，せめて，レーザー光線を吐くやつ(メデューサヘッドのことだね)だけでもつけたかった」

なるほど，これってあのマーベルランドを目指していたんですね。たすけて，パコーってか（ううっ，メガドラ版でもいってほしかった。ルクシーちゃん)。

市販のゲームとか，ゲームセンターのゲームとかを見てるとあんまりにもすごいから作るのめげてしまいそうだけど，でも，がんばっていけばきっといいものができますよ。プロの人だって最初は素人だったんだし，がんばればきっといつかマーベルランドだって，ファイナルファイトだっていけますよ。目指せ，ゲームデザイナー！

というわけで，わたしゃ，マーベルランドをまたやるのさ。へっへっへっ。いま，ワールド4の1だから，あと6面でゴールなのさ。今週中に行けるといいな。ああ，原稿は遅れるし，貧乏にはなるし罪なゲームだよ，まったく。

じゃ，また来月。このOh!Xで！

リスト1　SPRN.C

```
  1: /**********************************************************
  2:
  3:                      縮小印刷プログラム
  4:                                     (CZ-8PC3用)
  5:
  6:                                          By 野崎 哲也
  7:
  8: **********************************************************/
  9: #include <stdio.h>
 10: #include <iocslib.h>
 11: #include <doslib.h>
 12: #include <string.h>
 13:
 14: int moji,file,prin=0,big=0;
 15: int han16();
 16: int han24();
 17: int zen16();
 18: int zen24();
 19: int (*han)();
 20: int (*zen)();
 21:
 22: struct FNTBUF buff,*buf = &buff;
 23:
 24:.void main(argc,argv)
 25: int argc;
 26: unsigned char *argv[];
 27: {
 28:     int code,i,name;
 29:     han=han16;
 30:     zen=zen16;
 31:     name=se(argc,argv);
 32:     if ( (file=OPEN(argv[name],0)) < 0 ) {
 33:                     printf("ファイルが見つかりません。¥n");
 34:                     exit();
 35:     }
 36:     ini();
 37:     while(1) {
 38:             if ( (code=FGETC(file)) == EOF ) break;
 39:             if (code == 0x0d) {
 40:                     FGETC(file);
 41:                     cr();
 42:                     continue;
 43:             }
 44:             if (code == 9) {
 45:                             tab();
 46:                             continue;
 47:             }
 48:             if ( (0x80 <= code) && (code <= 0x9f) ){
 49:                                     (*zen)(code);
 50:                                     continue;
 51:             }
 52:             if (code <= 0x1f) continue;
 53:             if (0xe0 <= code) {
 54:                         (*zen)(code);
 55:                         continue;
 56:             }
 57:             (*han)(code);
 58:
 59:     cr();
 60:     CLOSE(file);
 61: }
 62:
 63: /*************** プリンター 初期化 ******************/
 64: ini()
 65: {
 66:     OUTLPT(27);
 67:     OUTLPT(99);
 68:     OUTLPT(49);                          /* リセット プリンター */
 69:
 70:     OUTLPT(27);
 71:     OUTLPT(37);
 72:     OUTLPT(57);
 73: if (big==0)    OUTLPT(12); /* 16ドットの時、12/120インチ紙送り量
の設定 */
 74:       else    OUTLPT(17); /* 24ドットの時、17/120インチ紙送り量
の設定 */
 75: }
 76:
 77: /********************** 改行 **************************/
 78: cr()
 79: {
 80:     if (prin) printf("¥n");
 81:     moji=0;
 82:     OUTLPT(10);                          /* 改行 */
 83: }
 84:
 85: /************** 24ドット・ビットイメージ ***************/
 86: imeg(n)
 87: char n;
 88:     {
 89:     OUTLPT(27);
 90:     OUTLPT(74);
 91:     OUTLPT(0);
 92:     OUTLPT(n);                       /* 24ﾄﾞｯﾄ･ｲﾒｰｼﾞ n行 */
 93:     }
 94:
 95: /**************** 16ﾄﾞｯﾄ 半角 ***************************/
 96: int han16(code)
 97: int code;
 98: {
 99:     register int i;
100:
101:     mojitest(9);
102:     if (prin) putchar(code);
103:     FNTGET(8,code,buf);
104:     imeg(9);
105:     for (i=7;i>=0;i--){
106:             OUTLPT(0);
107:             OUTLPT(hg(0,i)*128+hg(1,i)*64+hg(2,i)*32+hg
(3,i)*16+hg(4,i)*8+hg(5,i)*4+hg(6,i)*2+hg(7,i));
108:             OUTLPT(hg(8,i)*128+hg(9,i)*64+hg(10,i)*32+h
g(11,i)*16+hg(12,i)*8+hg(13,i)*4+hg(14,i)*2+hg(15,i));
109:     }
110:     OUTLPT(0);
111:     OUTLPT(0);
112:     OUTLPT(0);
113: }
```

```
114:
115: /***************  16ﾄﾞｯﾄ 全角  ******************/
116: int zen16(code1)
117: int code1;
118: {
119:      register int i;
120:      int code2;
121:
122:      if ( (code2=FGETC(file)) == EOF ) return(0);
123:      mojitest(18);
124:      if (prin) {
125:           putchar(code1);
126:           putchar(code2);
127:      }
128:
129:      FNTGET(8,(code1*256+code2),buf);
130:      imeg(18);
131:
132:      OUTLPT(0);
133:      OUTLPT(0);
134:      OUTLPT(0);
135:      for (i=7;i>=0;i--){
136:                OUTLPT(0);
137:                OUTLPT(hg(0,i)*128+hg(2,i)*64+hg(4,i)*32+hg
(6,i)*16+hg(8,i)*8+hg(10,i)*4+hg(12,i)*2+hg(14,i));
138:                OUTLPT(hg(16,i)*128+hg(18,i)*64+hg(20,i)*32
+hg(22,i)*16+hg(24,i)*8+hg(26,i)*4+hg(28,i)*2+hg(30,i));
139:      }
140:      for (i=7;i>=0;i--){
141:                OUTLPT(0);
142:                OUTLPT(hg(1,i)*128+hg(3,i)*64+hg(5,i)*32+hg
(7,i)*16+hg(9,i)*8+hg(11,i)*4+hg(13,i)*2+hg(15,i));
143:                OUTLPT(hg(17,i)*128+hg(19,i)*64+hg(21,i)*32
+hg(23,i)*16+hg(25,i)*8+hg(27,i)*4+hg(29,i)*2+hg(31,i));
144:      }
145:      OUTLPT(0);
146:      OUTLPT(0);
147:      OUTLPT(0);
148: }
149:
150: /*************  ビット単位の摘出  ********************/
151: hg(co,i)
152: int co;
153: int i;
154: {
155:       static power[]={1,2,4,8,16,32,64,128,256};
156:       register unsigned char c;
157:       c=buf->buffer[co];
158:       c=((c & power[i]) == power[i]);
159:       return(c);
160:  }
161:
162: /********************  TAB処理  ******************/
163: tab()
164: {
165:  do (*han)(' ');
166:  while (moji % 8);          /* ここの数字がTABが作る空白の字数 */
167: }
168:
169: /************  紙をはみ出す時の処理 **************/
170: mojitest(plus;
171: int plus;
172: {
173:     if ((moji+plus) > 1440) {  /* 1行に表示出来るドット数 */
174:                cr();
175:                moji=0;
176:     }
177:     else moji+=plus;
178: }
179:
180: /***************  24ﾄﾞｯﾄ 半角  ************************/
181: int han24(code)
182: int code;
183: {
184:      register int i;
185:
186:      mojitest(13);
187:      if (prin) putchar(code);
188:      FNTGET(12,code,buf);
189:      imeg(13);
190:
191:      for (i=7;i>=0;i--){
192:                OUTLPT(hg(0,i)*128+hg(2,i)*64+hg(4,i)*32+hg
(6,i)*16+hg(8,i)*8+hg(10,i)*4+hg(12,i)*2+hg(14,i));
193:                OUTLPT(hg(16,i)*128+hg(18,i)*64+hg(20,i)*32
+hg(22,i)*16+hg(24,i)*8+hg(26,i)*4+hg(28,i)*2+hg(30,i));
194:                OUTLPT(hg(32,i)*128+hg(34,i)*64+hg(36,i)*32
+hg(38,i)*16+hg(40,i)*8+hg(42,i)*4+hg(44,i)*2+hg(46,i));
195:      }
196:      for (i=7;i>=3;i--){
197:                OUTLPT(hg(1,i)*128+hg(3,i)*64+hg(5,i)*32+hg
(7,i)*16+hg(9,i)*8+hg(11,i)*4+hg(13,i)*2+hg(15,i));
198:                OUTLPT(hg(17,i)*128+hg(19,i)*64+hg(21,i)*32
+hg(23,i)*16+hg(25,i)*8+hg(27,i)*4+hg(29,i)*2+hg(31,i));
199:                OUTLPT(hg(33,i)*128+hg(35,i)*64+hg(37,i)*32
+hg(39,i)*16+hg(41,i)*8+hg(43,i)*4+hg(45,i)*2+hg(47,i));
200:      }
201:      OUTLPT(0);
202:      OUTLPT(0);
203:      OUTLPT(0);
204: }
205:
206: /***************  24ﾄﾞｯﾄ 全角  ******************/
207: int zen24(code1)
208: int code1;
209: {
210:      register int i;
211:      int code2;
212:
213:      if ( (code2=FGETC(file)) == EOF ) return(0);
214:      mojitest(26);
215:      if (prin) {
216:          putchar(code1);
217:          putchar(code2);
218:      }
219:
220:      FNTGET(12,(code1*256+code2),buf);
221:      imeg(26);
222:
223:      OUTLPT(0);
224:      OUTLPT(0);
225:      OUTLPT(0);
226:      for (i=7;i>=0;i--){
227:                OUTLPT(hg(0,i)*128+hg(3,i)*64+hg(6,i)*32+hg
(9,i)*16+hg(12,i)*8+hg(15,i)*4+hg(18,i)*2+hg(21,i));
228:                OUTLPT(hg(24,i)*128+hg(27,i)*64+hg(30,i)*32
+hg(33,i)*16+hg(36,i)*8+hg(39,i)*4+hg(42,i)*2+hg(45,i));
229:                OUTLPT(hg(48,i)*128+hg(51,i)*64+hg(54,i)*32
+hg(57,i)*16+hg(60,i)*8+hg(63,i)*4+hg(66,i)*2+hg(69,i));
230:      }
231:      for (i=7;i>=0;i--){
232:                OUTLPT(hg(1,i)*128+hg(4,i)*64+hg(7,i)*32+hg
(10,i)*16+hg(13,i)*8+hg(16,i)*4+hg(19,i)*2+hg(22,i));
233:                OUTLPT(hg(25,i)*128+hg(28,i)*64+hg(31,i)*32
+hg(34,i)*16+hg(37,i)*8+hg(40,i)*4+hg(43,i)*2+hg(46,i));
234:                OUTLPT(hg(49,i)*128+hg(52,i)*64+hg(55,i)*32
+hg(58,i)*16+hg(61,i)*8+hg(64,i)*4+hg(67,i)*2+hg(70,i));
235:      }
236:      for (i=7;i>=0;i--){
237:                OUTLPT(hg(2,i)*128+hg(5,i)*64+hg(8,i)*32+hg
(11,i)*16+hg(14,i)*8+hg(17,i)*4+hg(20,i)*2+hg(23,i));
238:                OUTLPT(hg(26,i)*128+hg(29,i)*64+hg(32,i)*32
+hg(35,i)*16+hg(38,i)*8+hg(41,i)*4+hg(44,i)*2+hg(47,i));
239:                OUTLPT(hg(50,i)*128+hg(53,i)*64+hg(56,i)*32
+hg(59,i)*16+hg(62,i)*8+hg(65,i)*4+hg(68,i)*2+hg(71,i));
240:      }
241:      OUTLPT(0);
242:      OUTLPT(0);
243:      OUTLPT(0);
244: }
245:
246: /*****************  スイッチ判定  ********************/
247: se(cc,vv)
248: int cc;
249: unsigned char *vv[];
250: {
251:      int i,chec=0,name=0;
252:
253:      for(i=1;i<cc;i++){
254:           chec=1;
255:           if (*(vv[i])=='/') {
256:                switch (*(vv[i]+1)){
257:                   case 'p':
258:                   case 'P':
259:                               prin=1;
260:                               break;
261:                   case 'b':
262:                   case 'B':
263:                               big=1;
264:                               han=han24;
265:                               zen=zen24;
266:                               break;
267:                   default:
268:                               printf("書式が間違っています。¥n");
269:                               exit();
270:               }
271:           }
272:           else name=i;
273:      }
274:      if (chec==0) {
275:                     printf("縮小印字プログラム  Ver 1.00     By
TESUYA NOZAKI¥n¥n");
276:                     printf("       書式  SPRN [/P][/B]  フ
ァイル名¥n¥n");
277:                     printf("    /P  ・・・・   画面にファイルの内
容を表示する。¥n");
278:                     printf("    /B  ・・・・   24ドット(指定され
ていなければ16ドット)で印刷する。¥n¥n");
279:                     exit();
280:      }
281:      if (name==0)  {
282:                      printf("ファイルが指定されていません。¥n");
283:                      exit();
284:      }
285:      return(name);
286: }
```

リスト2 MERVEL.BAS

```
10 COLOR7:WIDTH 40
20 LOCATE  8,10
30 PRINT"***   MERVEL X1   ***"
40 LOCATE 10,18
50  PRINT"push space key ..."
60 KEY 0,""
70  s$="":WHILE(s$="")
80  s$=INKEY$
90  WEND
100 KEY 0,""
110 '
120 COLOR 7
130 kankaku=10:dir =0
140 brank=10
150 LOx=20
160 sc=0:i=0
170 WIDTH 40:
180 LOCATE LOx,5:PRINT"@"
190 ' main
200 FOR y=0 TO 24
210    LOCATE  8,y
220   PRINT"|                        |";
230 NEXT
240 LOCATE LOx,5:PRINT"@"
250 BEEP:st$=SCRN$(LOx,5,1)
260 IF st$<>"@" THEN STOP
270 LOCATE LOx,5:PRINT"@"
280 a$=INKEY$(0)
290 'PRINT "i=";i
295 IF a$="5" THEN dir=5
300 IF a$="6" THEN dir=6
305 IF dir=6  THEN LOx=LOx+1
310 IF lox > 28 THEN lox =28
315 IF a$="4" THEN dir =4
```

```
320 IF dir=4  THEN LOx=LOx-1
330 IF lox <  9 THEN lox = 9
340       st$=SCRN$(LOx,5,1)
350  IF st$="*" THEN GOTO580
360 LOCATE LOx,5:PRINT"@"
370 GOTO  400
380 KEY 0,""
390 GOTO 280
400 '
410     LOCATE  8,24
420    PRINT"|                    |"
430 i=i+1:j=i MOD kankaku
440 'PRINT j ,i,kankaku
450 IF j=1 THEN sc=sc+1:GOTO 480
460 '      PRINT j;" : "
470 GOTO 380
480 LOCATE 8,23
490 PRINT"|********************|"
500 x=INT(RND(1)*(19-brank))+9
510 LOCATE x,23
520 FOR k=1 TO brank:PRINT" ";:NEXT
530 IF (sc MOD 16)=7 THEN kankaku=kankaku-1:i = kankaku+2
540 IF kankaku<4 THEN kankaku =3
550 IF (sc MOD 16)=15 THEN brank=brank-1
560 IF brank<3 THEN brank =2
570 GOTO 460
580 '---- game over ----
590 lx=lox:ly=5
600 COLOR 2:FOR i=0 TO 5
610 FOR WA =0 TO 100:NEXT
620 LOCATE lx+i,5:PRINT"*";
630 LOCATE lx+i,5+i:PRINT"*";
640 LOCATE lx-i,5:PRINT"*";
650 LOCATE lx+i,5-i:PRINT"*";
660 LOCATE lx,5+i:PRINT"*";
670 LOCATE lx-i,5+i:PRINT"*";
680 LOCATE lx,5-i:PRINT"*";
690 LOCATE lx-i,5-i:PRINT"*";
700 NEXT
710 COLOR 7:FOR i=0 TO 5
720 FOR WA =0 TO 100:NEXT
730 LOCATE lx+i,5:PRINT" ";
740 LOCATE lx+i,5+i:PRINT" ";
750 LOCATE lx-i,5:PRINT" ";
760 LOCATE lx+i,5-i:PRINT" ";
770 LOCATE lx,5+i:PRINT" ";
780 LOCATE lx-i,5+i:PRINT" ";
790 LOCATE lx,5-i:PRINT" ";
800 LOCATE lx-i,5-i:PRINT" ";
810 NEXT
820 CLS:LOCATE 10,10
825 KEY 0,""
830 PRINT"your score is ";sc
840 LOCATE 10,18
850 PRINT"push space key ..."
860 s$="":WHILE(s$="")
870 s$=INKEY$
880 WEND
890 GOTO 10
```

(で)のぱーてぃハンズ第3部——(その3)

ベースもガンガン上がってきて，このゲームもいよいよカンジンカナメの思考ルーチンのお話になるわけです。

さあ，Oh!Xはちゃんと買ってきたか？　心の準備はいいか？　立ち読みはあかんぞ！　夢に出るぞ！

第1のコース

さてさて，コンピュータにゲームをやらせるわけですが，コンピュータにどんなふうに“考え”させるんでしょう。

当然コンピュータが勝手に考えてくれるわけはない。んで，そこはやっぱり，ほかのプログラムと同じようにプログラムを組んでやって，あたかも考えているように見せる必要があるのですね。では，いったいどんなプログラムを作れば，そういうふうに見えるんでしょうか。

まず考えつくのが，“私だったらこうやるよん”方式のプログラムにする方法ですね。

まず，配列を用意します。その配列に“えっと，ここに相手がこれを打ってきたら，自分はここにこれを打つんだけどな”と思ったものをかたっぱしから，ダーッ，と書いておくわけです。で，コンピュータの思考ルーチンはそれをアンチョコにして，“あ，このパターンはあるな。じゃあ，ここに打とう”と，1つひとつ照らし合わせて打つ，というわけです。

もちろん，これならこれでもかまわないんですけどね。ただ，これだとなにしろできるルーチンが，自分の頭の質の落ちたコピー（だって，自分の考えを全部完璧に書きつくせるわけないもんね。何かひとつくらい絶対抜けちゃうでしょ？）になっちゃうから，あんまり面白くないし，それにちっとも強くない，というか少なくとも自分よりは絶対弱い。知識を詰め込んでおけば，それを勝手に解釈して自分でどんどん強くなっていってくれれば，もう本当にばんばんざいなんだけど，それをやるには本当に人工知能が必要になっちゃう。

というわけで，このテのプログラムを組むときは次の解説する第2の方法，“しらみつぶしに手を考えちゃうよん”方式のプログラムが多かったりするのです。

第2のコース

まず，コンピュータの苦手なこと。どんなものが思い浮かぶかな。まあ，普通に考えるとたいていアナログ的なものになりますよね。「だいたいこんなもんかなと考えてみる」，なんていうのは0か1かをバシッと決めるのがモットーのコンピュータ君には最大級に苦手なものなのであります。

一方，コンピュータの得意なことってどんなことがあるかっていうと，“ひたすら計算する”ことですね。そりゃあ，そうだ，だいたいコンピュータのことを電子計算機，なんて石器時代の昔にはいっていたとかいう話があるくらいなんだから（いま，コンピュータのことを電子計算機，なんていったら本当に石器時代人扱いされちゃうし）。たとえば，255＋192が384より大きいかどうか，なんてのを何千回，何万回とビュンビュン計算するなんてのは人間の何千倍も速いし，バグさえなければとっても正確にやってくれるのであります。

じゃあ，その得意な部分を生かしてやればいいじゃないか，得意な部分を伸ばそうと，まるで子供の教育みたいな考え方をいくのがさっきの“しらみつぶしに手を考えちゃうよん”方式のプログラムなのです。

ではここで質問。

問　コンピュータにゲームの手を打たせるようにします。いま，打てる手は0番〜2番までとします。ではあなたはコンピュータに何番の手を打たせるようにしますか？

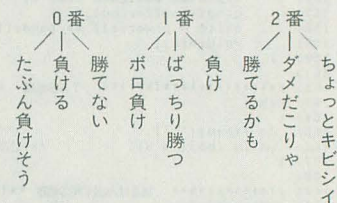
答　0番〜2番の中でいちばん勝ちに結びつきそうな手を打たせる

なんだよ，当たり前じゃないかって。うん。でも前の方法だと，たとえばこういう状況なら必ず1番，とコンピュータのアンチョコに書いてあったわけ。そこが違うところなのですね。

では第2問。

問　では，0番から2番の中でいちばん勝ちに結びつきそうな手はどれか？

答　それぞれの手の次の手を打ったときに，いちばん勝ちに結びつきそうな手を打つことができる手

なんかややこしいけどね。図にするとこういうことなのです。

上が先に打つ手。で，その手を打ったとき，次に選べる3つの手が勝てそうか，そうでないかが書いてあります。では，どの手を打つと勝てるでしょう。もちろん，1番を打ってから，次にばっちり勝つ手を打てばいいんですよね。

で，また問。

問　では次の手の中でいちばんいい手はどうやって判定するのか？

答　次の次の手が勝てそうな手になっている手

なんだか，禅問答みたいになってきてるけど，そう，その先の手を見てやってそれが勝てそうかどうか見てやればいいんです。ずっと，その調子で調べていくと……，そう，いつかはゲームオーバーになる手が出てきますよね。その手を打ったときに勝てるか負けるかは簡単にわかるでしょ。つまり，いま打てる手でいちばんいい手がどれかを考えるのに，次の手を打った場合，その次……と打てる手を全部調べてやればいいわけなんですね。これが第2の方法，“しらみつぶしに手を考えちゃうよん”方式のプログラムの基本的な考え方なんです。

たとえば，3択のゲームでも20手で終わるゲームだと，3^20=3486784401手（あはは，X68000の電卓機能はこういうとき便利だな）もあるので，ふつうの人間には絶対できない（一生かかってもゲームが終わらんわ）けど，コンピュータの素早さと正確さがあればなんとなくできそうな気がするでしょ。実際のプログラムでは時間がもったいないので，さらにちょっとしたことをするんだけど，それはあとでね。

というわけで，今月は思考ルーチンの基本的な考え方のお話でした。リストはなし。では来月，またこのOh!Xで。「でやっ！」と去る。

第110部 SLANG用NEWファイル入出力ライブラリ

●Cプログラマへの道

積み上げてきた数々の努力がやっと結実した感のあるCコンパイラですが，皆さんいかがお使いでしょうか。そろそろ編集室にも皆さんからのお便りが届き始めており，動き出したなあという実感があります。システム記述言語として有名なC言語ですから，S-OSシステムをCで書き直した……という人も登場するかもしれませんね。

今回の移植で初めてC言語を使うという方も少なくないでしょう。そこで編集部では，10月号から2回にわたってSmall-Cのガイダンスを予定しています。今や標準プログラミング言語になった感のあるC言語。ぜひこの機会にチャレンジしてみてください。そして，チョットいいプログラムができたら編集部への投稿をお忘れなく。このところ大物の掲載が続いているからといって遠慮する必要はありません。皆さんのちょっとしたアイデアや思いつきが，S-OSの大きな前進になることもあるのです。

●SLANG用NEWファイル入出力ライブラリ

1988年10月号で発表されたSLANG用のファイル入出力ライブラリは，ファイルの先頭から順にデータの入出力を行うことしかできないS-OSの能力を拡張し，ファイル内の好きなデータを文字単位で取り出すことができるようにするものでした。ファイル内にある任意の位置のデータを読み書きするというこのような処理は，本来S-OSがサポートしているべき機能です。ただしこれをテープに対応させるのは難しいため，"SWORD"制作時には見送られたのでした。そのため，このファイル入出力ライブラリでも扱えるファイルはディスク上のファイルのみとなっています(RAMディスク上のファイルでも可)。ダンプリストはSLANGのランタイムルーチンに付け加える形で発表されましたが，SLANGには依存していないので別の言語に組み込んで使うこともできます。必要に応じて各自が使用するという形態がとられたのです。

ファイルI/Oがサポートされてますます C言語に近くなったSLANGですが，さらに1989年12月号で，I/Oリダイレクションまでもサポートされ，ますます活動の可能性を拡げていったのは記憶に新しいところです。

しかし，このファイル入出力ライブラリには若干の問題点がありました。今回発表するファイル入出力ライブラリはこの問題点を解決しつつ機能を拡張したもので，より柔軟に，より自由にファイルを扱うことができるようになっています。ソースリストが公開されており，だれもが自由に改良・変更することができるS-OSならではのメリットだといえるでしょう。ユーザーの提言がユーザーによって吟味され新しい提言へとつながる。1988年12月号のファイル入出力ライブラリから今回のNEWライブラリにつながる流れを見ていると，S-OSの基調であるこの精神を改めて実感します。

●S-OSの系譜(24)

1987年11月号のSENTINELは「全機種共通システムS-OS再考」と題された特集でした。特集は泉大介氏の「神話の中のマイクロコンピュータ」という序論で始まります。「私たちはコンピュータを使う多くの友を大切にしたい。X1ユーザーの作り出すプログラムを，自分はMZユーザーだからなんてケチな理由で拒絶したくない。PCユーザーの頭脳を，シャープユーザーだからなんてつまらない理由で拒否したくない」と，S-OSの精神・心構えといったものが"SWORD"の開発に携わった同氏の雅語を交じえた文体で語られています。

続いてこれまでに発表された数々のプログラムの概要が「S-OSの仲間たち」というタイトルで掲載されています。「もうひとつのFuzzyBASIC入門」では，高級言語性と低級言語性を兼ね備えたFuzzyBASICの開発に込めた想いが，作者の瀧山氏によって語られました。アブソリュートアセンブラしかないS-OSで，リロケータブルオブジェクトを扱う方法について田島氏からの提言があり，CARDDRVでお馴染みの毛内氏は「S-OSこちら集中治療室」で，S-OSとマシン語を扱うためのガイドをしています。

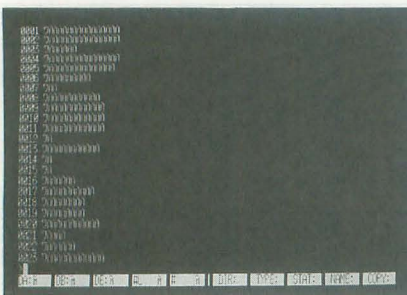
挑戦的な話題，入門しようという人へのガイドなどが随所に設けられた囲みで取り上げられ，高度成長期にあったS-OSとそれを支えるスタッフの意気込みが紙面から伝わってくるようでした。熱っぽい日々，溢れ出るアイデア，挑戦と挫折と克服。そのすべてがこの特集に集約されています。

アミューズメントプログラムとしては，BACK GAMMONが発表されました。Fuzzy BASICとコンパイラによって開発されたこのプログラムは，インタプリタでの開発・デバッグとコンパイラによる高速化の格好の実例となりました。

全機種共通
S-OS"SWORD"要

NEWファイル入出力ライブラリ
SLANG

Itou Masahiko
伊藤 雅彦

1988年10月号のファイル入出力ライブラリが機能拡張され，さらに使いやすくなりました。これを使えばSLANGで簡単にファイル処理を行うことができます。アプリケーション作成に活用してください。

Oh!X 1988年10月号でSLANG用ファイル入出力ライブラリが発表されました。先頃，私もこれを使ってプログラムを作ろうと思いちょっと使ってみると自分の思っているような動作をしてくれない。そこで，調べてみたら次のような問題点が次々と発見されました。

1) IYレジスタを破壊する（SLANGでは動的変数のポインタとしてIYレジスタを使っているから，IYレジスタを破壊するようなプログラムはSLANG用関数とはいえない）

2) ディレクトリ上のファイルサイズのデータを管理していない（そういう仕様なのでバグではないが，作成したファイルをエディタなどで読み込めないため用途が格段に狭くなる）

3) 同じデバイスの2つのファイルを同時に書き込みオープンして書き込みを行うと，FATの管理ができずに間抜けなことをする（2つのファイルについて別々にFATを管理しているため）

4) ひとつのファイルを2つのファイルナンバーでオープンするとうまく動作しない（このような事態を想定していないため）

2)～4)についてはそういう仕様だといい切れるかもしれませんが，やっぱりプログラムを作るうえで問題となります。確かにそういうものだと納得するのもいいですが僕は思い切ってこれらの問題点を解消して，さらに機能拡張した新関数を作成してしまいました。ただし4)については，すべて読み込みオープンのみの場合だけサポートしています。

拡張された新関数の機能

使い方は旧関数と同じで，拡張した機能は以下のとおりとなっています。

1) ファイルナンバーを0～3の4つに増やした。

2) FSEEK関数以外では，HLレジスタだけでエラーチェックまでできる（HLレジスタの内容はSLANG関数の戻り値になるためエラーチェックまでできると便利）

3) FSEEK関数のシークモードにファイルの終端を基準とするモードを追加した

4) セクタ単位のほかにクラスタ単位でディスクアクセスをすることができる

5) ディレクトリ上のファイルサイズのデータが0000Hであるファイル（つまり旧関数で新規作成されたファイル）をオープンした場合，FATから大雑把なファイルサイズを割り出して管理を行う

5)についてはファイルサイズ管理をきちんとしたため旧関数で作成したファイルを正常に扱うことができなくなってしまったのです。しかし，それでは旧関数の上位互換を名乗れないので急きょ付け加えました。これで旧関数で作成したファイルもかなり救われますが，ほとんどの場合はファイルに余計なデータがくっついた格好になります。またエディタなどで作成し，旧関数でデータを追加したファイルは正常に扱えません。この場合にはディスクエディタなどを使ってディレクトリ上のデータを修正してください。

機能一覧

では，それぞれの関数の機能を紹介していきます。アドレスはライブラリ先頭アドレスからのオフセットです。

・FOPEN(FNO, FNAME, MODE)

機能：ファイルをオープンする
アドレス：0048_H
　FNO (HL)：ファイルナンバー
　FNAME (DE)：ファイル名が格納されている先頭アドレス
　MODE (BC)：オープンモード
　　0：読み込み専用
　　1：書き込み専用
　　2：読み書き両用
　　3：新規オープン
戻り値
正常終了：Cy=0, A=0
　　　　　H=0, L=0
エラー　：CY=1, A=エラーナンバー
　　　　　$H=FF_H$, L=エラーナンバー

・FSEEK(FNO, OS, MODE)

機能：ファイルポインタを移動する
アドレス：$01C5_H$
　FNO (HL)：ファイルナンバー
　OS (DE)：ファイルポインタの移動位置のオフセット
　MODE (BC)：移動位置の基準
　　0：ファイルの先頭(OSの値は0～65535とみなされる)
　　1：現在のファイルポインタの位置(OSの値は－32768～32767とみなされる)
　　2：ファイルの終端(OSの値は－65535～0とみなされる)
戻り値
正常終了：Cy=0, A=0
　　　　　HL= 移動後のファイルポインタの位置
エラー　：Cy=1, A=エラーナンバー

HL＝FFFF$_H$

・FPUTC(FNO, CHR)

機能：ファイルに1バイトデータを書き込む

アドレス：0216$_H$

FNO（HL）：ファイルナンバー

CHR（DE）：書き込む1バイトデータ

戻り値

正常終了：Cy＝0, A＝0

H＝0, L＝0

エラー　：Cy＝1, A＝エラーナンバー

H＝FF$_H$, L＝エラーナンバー

・FGETC(FNO)

機能：ファイルから1バイトデータを読み込む

アドレス：0225$_H$

FNO（HL）：ファイルナンバー

戻り値

正常終了：Cy＝0, A＝0

H＝0, L＝読み込んだデータ

エラー　：Cy＝1, A＝エラーナンバー

H＝FF$_H$, L＝エラーナンバー

・FCLOSE(FNO)

機能：ファイルをクローズする

アドレス：030F$_H$

FNO（HL）：ファイルナンバー

戻り値

正常終了：Cy＝0, A＝0

H＝0, L＝0

エラー　：Cy＝1, A＝エラーナンバー

H＝FF$_H$, L＝エラーナンバー

エラーナンバー&ワークエリア

次はエラーナンバーについて説明します。関数でエラーが発生したときにエラーナンバーを返しますが, S-OSで決められているエラー番号1～14のほかに15～20の番号を返す場合があります。それらの意味は以下のとおりです。

15 Bad Device

デバイスがディスクではない

16 Open Mode Error

読み込みモードでオープンしたファイルに書き込みを行おうとした。あるいは書き込みモードでオープンしたファイルから読み込みを行おうとした

17 Out of File

ファイルポインタがファイル外になってしまった

18 Already Open

オープンしようとしたファイルナンバーがすでにオープン中のファイルに存在する

19 Double Open

オープンしようとしたファイルがすでにほかのファイルナンバーでオープンされている

20 Too Long File

ファイルサイズが長すぎる(65535バイトを超えている)

ワークエリアについては表1にまとめておきました。これらのワークエリアはファイルオープン中に書き換えてはいけませんので注意してください。

ライブラリを組み込む

これらの関数を使用するためには, SLANGからMACHINE関数を使って外部関数として登録するか, SLANG自体にライブラリを登録してやる必要があります。MACHINE関数を使用した場合には, プログラムごとに関数を登録する必要があるのでちょっと面倒です。ここではライブラリの追加をするという方法を書きます。組み込み手順は,

1) リスト1のファイル入出力ライブラリをSLANGのランタイムルーチンの最後に追加する

2) ランタイムルーチンの最終アドレス(730C$_H$)に300A$_H$を登録

3) SLANG本体の67AF$_H$からの組み込み関数テーブルに関数を登録(リスト2の変更点を加える)

4) 66A1$_H$からのデータ領域にデータエリアを登録(66AD$_H$以降の4バイトを80$_H$ 6D$_H$ C7$_H$ 6D$_H$とする)

サンプル1 ワハハ

```
 1  //
 2  //  ﾜﾊﾊ ﾜﾊﾊﾊ ﾜﾊ ﾜﾊ
 3  //                    (ats)
 4  //
 5  //  WORK  $3000  ;  / For MZ-80
 6 MAIN()
 7  BEGIN
 8   FOPEN(0,"ﾜﾊﾊ",3);
 9     FILL();
10     SKIP();
11   FSEEK(0,2,2);
12   FPUTC(0,'¥N');
13   FPUTC(0,0);
14   FCLOSE(0);
15  END;
16 FILL()
17   VAR  I;
18  BEGIN
19   FPUTC(0,'ﾜ');
20    FOR  I=1  TO  $1FE [
21     FPUTC(0,'ﾊ');
22    ]NEXT;
23  END;
24 SKIP()
25   VAR  A;
26  BEGIN
27   FSEEK(0,16,0);
28   A=16;
29    REPEAT [
30     FSEEK(0,A,0);
31      FPUTC(0,'¥N');
32      FPUTC(0,'ﾜ');
33     A=A+RND(16)+3;
34    ]UNTIL A>$1F8;
35  END;
```

表1 ワークエリア

0000$_H$ ：ディスクアクセスをセクタ単位で行うかクラスタ単位で行うかの指定（初期値は1）

0：セクタ単位（ランダムっぽいアクセス向き）

1：クラスタ単位（シーケンシャルっぽいアクセス向き）

0001$_H$ ：FATバッファの先頭アドレス（下位）

0002$_H$ ：FATバッファの先頭アドレス（上位）

このアドレスから400$_H$バイトをFATバッファとして使用。ただし, 使用するデバイスが3つ以下の場合には, デバイスの数×100$_H$バイトを使用

0003$_H$ ：ファイルアクセス用のデータバッファの先頭アドレス（下位）

0004$_H$ ：ファイルアクセス用のデータバッファの先頭アドレス（上位）ディスクアクセスをセクタ単位で行う場合は, このアドレスから400$_H$バイトを使用。クラスタ単位の場合は4000$_H$バイトを使用

FATバッファの初期値はC800$_H$, データバッファの初期値はCC00$_H$になっています

SLANGのバグ情報

見つかったバグは2つ, 割り算ルーチンとSGN関数です。まず, 割り算ルーチンについてですが試しに以下のリストを入力してコンパイル, 実行してみてください。

```
MAIN()
  BEGIN
      PRINT(32768/255,/)
  END
```

すると"128"と表示されるはずなのですが"0"と表示されてしまいます。原因は1988年3月号掲載のソースリスト, 153ページ, 642行目,

```
ADC  A, A
```

のところでAレジスタのキャリを考慮していないためです。デバッグはリスト3の訂正を加えてください。

SGN関数は関数の仕様自体に疑問があります。SLANGのソースリストでSGN関数は, 第7ビット目で正負を判断しようとしているのです。つまりSGN関数の引数は1バイトとみなしているわけです。

ところがSLANGでは通常2バイトで計算を行っていますから, SGN関数も2バイトの引数を数値として扱うのが自然ではないでしょうか。以上のような考えに基づいてSGN関数は第15ビット目で正負を判断するようにしました。こちらのデバッグはリスト4の訂正を加えてください。

5) クラスタバッファとテキスト先頭アドレスを7400_Hにする（$300D_H$からの4バイトを，00_H 74_H 00_H 74_Hとする）

となります。

以上の変更を行って先頭アドレスからランタイムルーチンの最終アドレスまでをSAVEすれば，ファイル入出力関数を組み込んだSLANGができあがります。

（編注）サンプル1は拡張関数のサンプルプログラムです。とりあえずコンパイルして実行すると“ワハハ”というテキストファイルが生成されます。内容はランダムな行数，ランダムな文字数で“ワハハハハ……”という文字列が並んでいるだけです。だからどうした，といわれればそれまでですが動作チェックに使ってください。

なお1989年12月号で紹介された，ファイルリダイレクションライブラリ「DIO.LIB」には対応していませんので注意してください。

参考文献

Oh!X 1988年10月号SENTINEL（華門真人）
Oh!MZ 1987年9月号 X68000 BASIC入門「激震のファイル入出力関数」（中森章）
Oh!MZ 1987年10月号 X68000 BASIC入門「追撃ランダムファイル」（中森章）
Oh!X 1990年8月号 X-BASICプログラミング調理実習「超入門・ファイル処理」（泉大介）
Oh!X 1990年9月号 X-BASICプログラミング調理実習「ファイルの魔術師fseek関数」（泉大介）

リスト1

```
6D80 00 00 C8 00 CC 00 00 00 : 94
6D88 00 00 00 00 00 00 00 00 : 00
6D90 00 00 00 00 00 00 00 00 : 00
6D98 00 00 00 00 00 00 01 00 : 01
6DA0 00 00 00 00 00 00 00 00 : 00
6DA8 00 00 00 00 02 00 00 00 : 02
6DB0 00 00 00 00 00 00 00 00 : 00
6DB8 00 00 03 00 00 00 00 00 : 03
6DC0 00 00 00 00 00 00 00 00 : 00
6DC8 CD DC 72 DA 08 73 3E 12 : C0
6DD0 C2 08 73 79 E6 FC B0 3E : 86
6DD8 0E C2 08 73 DD 71 02 3E : D9
6DE0 04 CD A3 1F DA 08 73 3A : 22
6DE8 5D 1F CD 63 28 4F 3E 0F : 70
6DF0 CA 08 73 DD 71 03 CD EB : 4E
6DF8 70 DA 08 73 DD 72 05 1C : 35
---------------------------------
SUM: 38 74 A3 98 E9 AC 74 DE 3760

6E00 1D 20 36 EB FD E5 FD 21 : 5E
6E08 90 6D 11 0E 00 06 04 FD : 23
6E10 7E 01 B7 28 1D FD 7E 03 : F9
6E18 DD BE 03 20 15 FD 7E 05 : 53
6E20 BC 20 0F FD 7E 02 DD B6 : FB
6E28 02 28 07 FD E1 3E 13 C3 : 23
6E30 08 73 FD 19 10 D9 FD E1 : 58
6E38 EB DD 7E 02 FE 03 CA C1 : D4
6E40 6E 1C 1D 4F 3E 08 C2 08 : 06
6E48 73 2A 74 1F 7E E6 07 FE : 99
6E50 04 C2 06 73 11 12 00 19 : 7B
6E58 5E 23 56 0C 0D 28 05 CB : E8
6E60 77 C2 02 73 DD 73 0C DD : E7
6E68 72 0D 01 0B 00 09 7E DD : EF
6E70 77 0A D5 CD F4 71 D1 DA : 33
6E78 08 73 7A B3 20 1E CD 84 : 37
---------------------------------
SUM: 64 5B D1 41 67 34 AA 43 FCFB

6E80 72 DA 3D 6F D6 7F 4F 78 : 14
6E88 FE 10 3E 14 D2 3D 6F 78 : 56
6E90 07 07 07 07 81 57 B7 3E : E9
6E98 14 CA 3D 6F DD 73 06 DD : BD
6EA0 72 07 AF CD 91 71 D4 00 : CB
6EA8 20 DA 3D 6F AF DD 77 04 : AD
6EB0 DD 77 08 DD 77 09 DD 77 : 0D
6EB8 0B DD 36 01 01 6F C3 FF : 51
6EC0 72 1D 28 32 2A 74 1F 3E : E4
6EC8 07 A6 FE 04 C2 06 73 CB : B5
6ED0 76 C2 02 73 11 1E 00 19 : F5
6ED8 5E DD 73 0A CD F4 71 DA : C4
6EE0 08 73 CD 84 72 DA 3D 6F : C4
6EE8 01 80 00 DD 7E 0A 60 6F : B5
6EF0 19 7E 70 B9 38 F8 CD F4 : B1
6EF8 71 DA 08 73 CD B6 72 38 : F3
---------------------------------
SUM: E5 9D C9 53 7D 6A 45 8B 41B7

6F00 3C 2A 74 1F 11 12 00 19 : 35
6F08 36 01 06 0B 23 72 10 FC : E9
6F10 23 77 DD 77 0A 0E 01 CD : D4
6F18 5F 71 38 21 EB 2A 74 1F : D1
6F20 01 20 00 ED B0 CD 5F 71 : 5B
6F28 38 13 DD 36 06 01 DD 36 : 78
6F30 07 00 DD 36 0C 01 DD 36 : 3A
6F38 0D 00 C3 AC 6E F5 CD 2E : DA
6F40 72 F1 C3 08 73 CD DC 72 : BC
6F48 DA 91 6F 3E 0C CA 91 6F : EE
6F50 3E 0E 04 05 C2 91 6F 0C : 23
6F58 0D 28 20 0D 28 0D 0D 20 : C4
6F60 30 DD 6E 06 DD 66 07 19 : E4
6F68 EB 18 10 DD 6E 08 DD 66 : A9
6F70 09 19 CB 7A 28 01 3F DA : A9
6F78 8F 6F EB DD 6E 06 DD 66 : 7D
---------------------------------
SUM: 8B 7B 96 59 A3 2A 54 D8 A0C4

6F80 07 AF ED 52 DA 8F 6F DD : AA
6F88 73 08 DD 72 09 EB C9 3E : C5
6F90 11 21 FF FF 37 C9 0E 00 : 3E
6F98 CD B0 6F 73 DD 36 04 01 : 77
6FA0 2E 00 C3 FF 72 16 00 0E : 86
6FA8 01 CD B0 6F 6E C3 FF 72 : 8E
6FB0 D5 CD DC 72 DA 8A 70 20 : E4
6FB8 05 3E 0C C3 8A 70 DD 7E : 67
6FC0 02 B9 3E 10 CA 8A 70 14 : E1
6FC8 15 28 05 3E 0E C3 8A 70 : 4B
6FD0 DD 5E 08 DD 7E 06 BB 20 : 7F
6FD8 45 DD 56 09 DD 7E 07 BA : 9D
6FE0 20 3C 0D 3E 11 CA 8A 70 : 7C
6FE8 7A A3 3C 3E 14 CA 8A 70 : 6F
6FF0 1C 1D 20 23 CD A6 72 7A : DB
6FF8 E6 0F CB FF 20 06 CD B6 : 68
---------------------------------
SUM: 36 87 68 AB 80 5D A5 A8 5115

7000 72 DA 8A 70 08 D5 CD 84 : 74
7008 72 D1 DA 8A 70 08 77 21 : B7
7010 89 6D 06 00 09 36 01 13 : 4F
7018 DD 73 06 DD 72 07 3A 80 : 66
7020 6D B7 DD 7E 09 28 02 E6 : 98
7028 F0 DD BE 0B 28 2E DD 7E : 47
7030 04 B7 28 10 DD 7E 0B CD : 26
7038 91 71 D4 03 20 DA 8A 70 : CD
7040 DD 36 04 00 DD 7E 09 CD : 48
7048 91 71 4F D4 00 20 DA 8A : A9
7050 70 DD 7E 09 0D 28 02 E6 : F1
7058 F0 DD 77 0B 2A 83 6D 3A : A3
7060 80 6D B7 DD 7E 00 28 0B : 32
7068 87 87 87 87 5F DD 7E 09 : DF
7070 E6 0F 83 84 67 DD 5E 08 : A6
7078 16 00 19 DD 5E 08 DD 56 : A5
---------------------------------
SUM: 0D AB 29 20 D7 D3 26 C2 3670

7080 09 13 DD 73 08 DD 72 09 : CC
7088 D1 C9 E1 E1 C3 08 73 CD : 67
7090 DC 72 DA 08 73 3E 0C CA : B7
7098 08 73 DD 7E 04 B7 28 0C : C5
70A0 DD 7E 0B CD 91 71 D4 03 : 0C
70A8 20 DA 08 73 DD 7E 06 DD : B3
70B0 BE 0C 20 08 DD 7E 07 DD : 31
70B8 BE 0D 28 1C 0E 01 CD 5F : 4A
70C0 71 DA 08 73 11 12 00 19 : 02
70C8 DD 7E 06 77 DD 7E 07 23 : 5D
70D0 77 4A CD 5F 71 DA 08 73 : B3
70D8 DD 36 01 00 CD 2E 72 2E : AF
70E0 00 D2 FF 72 DD 36 01 01 : 58
70E8 C3 08 73 3E FF 32 8D 6D : A7
70F0 0E 00 ED 5B 60 1F 21 10 : 06
70F8 00 19 22 8E 6D 2A 64 1F : E3
---------------------------------
SUM: AA FD 2D 20 70 91 5B 42 A6FF

7100 3E 01 CD 00 20 D8 06 08 : 12
7108 7E 3C 28 46 3D 20 0C 3A : CB
7110 8D 6D 3C 20 2A 79 32 8D : B8
7118 6D 18 24 E5 D5 C5 ED 5B : 70
7120 74 1F 06 10 23 13 1A BE : B7
7128 20 02 10 F8 C1 D1 E1 20 : BD
7130 0E 79 ED 5B 74 1F 01 20 : 83
7138 00 ED B0 57 58 B7 C9 0C : D8
7140 3E 20 85 6F 30 01 24 10 : B7
7148 BF 13 2A 8E 6D B7 ED 52 : ED
7150 20 AB 3A 8D 6D FE FF 20 : 1C
7158 01 79 57 1E 01 B7 C9 DD : 4D
7160 7E 03 32 5D 1F DD 7E 05 : 8F
7168 0F 0F 0F 47 E6 1F ED 5B : C1
7170 60 1F 83 5F 30 01 14 2A : D0
7178 64 1F 3E 01 0C 0D 20 04 : FF
---------------------------------
SUM: C7 F0 4A B1 58 67 6E 21 CCF8

7180 CD 03 20 C9 CD 00 20 D8 : 7E
7188 78 E6 E0 85 6F D0 24 B7 : DD
7190 C9 DD 5E 0A 57 E6 F0 28 : 63
7198 28 0F 0F 0F 0F F5 CD A6 : CC
71A0 72 79 CD D6 72 C1 4A 16 : 21
71A8 00 EB 19 7E B7 28 04 FE : 63
71B0 80 38 04 3E 07 37 C9 26 : 27
71B8 00 6F 10 EE 79 E6 0F 57 : 32
71C0 5D DD 7E 03 32 5D 1F 4A : B3
71C8 7B 07 07 07 07 5F E6 0F : EB
71D0 57 7B E6 F0 81 5F 2A 83 : 35
71D8 6D 3A 80 6D B7 DD 7E 00 : A6
71E0 20 05 84 67 AF 3C C9 87 : 4B
71E8 87 87 87 84 67 7B E6 F0 : D1
71F0 5F 3E 10 C9 CD A6 72 79 : D4
71F8 FE 04 3F D0 21 84 6D 0E : 31
---------------------------------
SUM: C8 47 AC D2 C0 8A 62 C8 64D0

7200 FF AF 23 0C BE 20 FB DD : 93
7208 7E 03 32 5D 1F 47 79 CD : BC
7210 D6 72 ED 5B 5E 1F 3E 01 : 4C
7218 CD 00 20 D8 21 85 6D 79 : 51
7220 85 6F 30 01 24 70 11 04 : CE
7228 00 19 36 00 B7 C9 DD 4E : FA
7230 03 DD E5 DD 21 90 6D 11 : D1
7238 0E 00 2E 01 06 04 DD 7E : A2
7240 01 B7 28 07 DD 7E 03 B9 : FE
7248 20 01 2C DD 19 10 EF DD : 1F
7250 E1 B7 2D C0 CD A6 72 21 : 8B
7258 89 6D 79 85 6F 30 01 24 : B8
7260 7E 11 04 00 B7 ED 52 36 : BF
7268 00 B7 C8 E5 DD 7E 03 32 : F4
7270 5D 1F 47 79 CD D6 72 ED : 3E
7278 5B 5E 1F 3E 01 CD 03 20 : 07
---------------------------------
SUM: 77 AA 07 40 F2 4A 86 55 B897

7280 E1 D0 70 C9 79 CD D6 72 : 78
7288 EB 26 00 DD 6E 0A 44 19 : C3
7290 7E B7 28 0E FE 80 30 06 : 1F
7298 04 6F 26 00 18 F1 FE 90 : 30
72A0 3F D0 3E 07 37 C9 21 85 : FA
72A8 6D 01 00 04 DD 7E 03 BE : 8E
72B0 C8 23 0C 10 FA C9 79 CD : 10
72B8 D6 72 AF 06 80 BE 28 07 : 6A
72C0 23 10 FA 3E 09 37 C9 36 : AA
72C8 80 3E 80 90 06 00 21 89 : 7E
72D0 6D 09 36 01 B7 C9 2A 81 : D8
72D8 6D 84 67 C9 7D E6 FC B4 : 34
72E0 3E 0E 37 C0 D5 11 0E 00 : 37
72E8 DD 21 90 6D 7D 0F 30 02 : B9
72F0 DD 19 0F 30 04 DD 19 DD : 0C
72F8 19 D1 DD 7E 01 B7 C9 AF : 75
---------------------------------
SUM: 26 76 81 48 25 B0 3D BA 3230

7300 67 C9 3E 04 18 02 3E 06 : D0
7308 26 FF 6F 37 C9          : 94
---------------------------------
SUM: 8D C8 AD 3B E1 02 3E 06 AA7A
```

リスト2 関数テーブルの追加

```
68C5 81 03 3A 04 46 4F 50 45 : EC
68CD 4E 0D 81 03 B7 05 46 53 : 34
68D5 45 45 4B 0D 81 02 08 06 : 73
68DD 46 50 55 54 43 0D 81 01 : 11
68E5 17 06 46 47 45 54 43 0D : 93
68ED 81 01 01 07 46 43 4C 4F : AE
68F5 53 45 0D 00 00 00 00 00 : A5
68FD 00 00 00 00 00 00 00 00 : 00
---------------------------------
SUM: 45 F1 AF B6 4C FA AE FB 6D5D
```

リスト3 割り算ルーチン訂正

```
6A3A 29 8F 38 03 BB 38 02 93 : 7B
6A42 2C 10 F5 5F C9 00 00 00 : 59
6A4A 00 00 00                : 00
---------------------------------
SUM: 55 9F 2D 62 84 38 02 93 9DD2
```

リスト4 SGN関数訂正

```
6BC4 7C B5 C8 CB 7C 21 01 00 : 62
---------------------------------
SUM: 7C B5 C8 CB 7C 21 01 00 DB00
```

リスト5

```
0000              1 ;
0000              2 ; S L A N G   F i l e   C o m m a n d
0000              3 ;
0000              4 ;   FOPEN(FNO,FNAME,MODE)
0000              5 ;   FSEEK(FNO,OS,MODE)
0000              6 ;   FPUTC(FNO,DATA)
0000              7 ;   FGETC(FNO)
0000              8 ;   FCLOSE(FNO)
0000              9 ;
0000             10 %TPCHK: EQU 02863H
0000             11 #FILE:  EQU 01FA3H
0000             12 #DRDSB: EQU 02000H
0000             13 #DWTSB: EQU 02003H
0000             14 ;
0000             15 #IBFAD: EQU 01F74H
0000             16 #DTBUF: EQU 01F64H
0000             17 #DIRPS: EQU 01F60H
0000             18 #FATPOS:EQU 01F5EH
0000             19 #DSK:   EQU 01F5DH
0000             20 ;
0000             21 ;
0000             22 BADDEVICE:     EQU 15
0000             23 OPENMODEERROR:EQU 16
0000             24 OUTOFFILE:     EQU 17
0000             25 ALREADYOPEN:   EQU 18
0000             26 DOUBLEOPEN:    EQU 19
0000             27 TOOLONGFILE:   EQU 20
0000             28 ;
0000             29 ;
0000             30 FNO:     EQU  0 ; 1 byte
0000             31 OPENFLG:EQU  1 ; 1
0000             32 MODE:    EQU  2 ; 1
0000             33 DEVICE: EQU  3 ; 1
0000             34 SCWFLG: EQU  4 ; 1
0000             35 DIRPT:  EQU  5 ; 1
0000             36 SIZE:   EQU  6 ; 2
0000             37 POINT:  EQU  8 ; 2
0000             38 CHEAD:  EQU 10 ; 1
0000             39 ACTSC:  EQU 11 ; 1
0000             40 DSIZE:  EQU 12 ; 2
0000             41 ;       EQU 14
0000             42 ;
0000             43 ;
0000             44 ;
0000             45  OFFSET 1000H
6D80             46  ORG    6D80H
6D80             47 ;
6D80             48 ;
6D80             49 CMODE:
6D80 00          50' DB 0
6D81             51 FBUFAD:
6D81 00 C8       52  DW 0C800H
6D83             53 SCBUFAD:
6D83 00 CC       54  DW 0CC00H
6D85             55 FBUFDEV:
6D85 00 00 00    56  DB 0,0,0,0
6D88 00
6D89             57 FBWRFLG:
6D89 00 00 00    58  DS 4
6D8C 00
6D8D             59 ;
6D8D             60 SPACEOS:
6D8D 00          61  DS 1
6D8E             62 DIREND:
6D8E 00 00       63  DS 2
6D90             64 ;
6D90             65 WORK0:
6D90 00 00       66  DB 0,0
6D92 00 00 00    67  DS 12
6D95 00 00 00
6D98 00 00 00
6D9B 00 00 00
6D9E             68 WORK1:
6D9E 01 00       69  DB 1,0
6DA0 00 00 00    70  DS 12
6DA3 00 00 00
6DA6 00 00 00
6DA9 00 00 00
6DAC             71 WORK2:
6DAC 02 00       72  DB 2,0
6DAE 00 00 00    73  DS 12
6DB1 00 00 00
6DB4 00 00 00
6DB7 00 00 00
6DBA             74 WORK3:
6DBA 03 00       75  DB 3,0
6DBC 00 00 00    76  DS 12
6DBF 00 00 00
6DC2 00 00 00
6DC5 00 00 00
6DC8             77 ;
6DC8             78 ;
6DC8             79 FOPEN:
6DC8 CD DC 72    80  CALL SETIX
6DCB DA 08 73    81  JP   C,ERROR
6DCE 3E 12       82  LD   A,ALREADYOPEN
6DD0 C2 08 73    83  JP   NZ,ERROR
6DD3 79          84  LD   A,C
6DD4 E6 FC       85  AND  0FCH
6DD6 B0          86  OR   B
6DD7 3E 0E       87  LD   A,14 ; bad data
6DD9 C2 08 73    88  JP   NZ,ERROR
6DDC DD 71 02    89  LD   (IX+MODE),C ; **MODE
6DDF 3E 04       90  LD   A,004H
6DE1 CD A3 1F    91  CALL #FILE
6DE4 DA 08 73    92  JP   C,ERROR
6DE7 3A 5D 1F    93  LD   A,(#DSK)
6DEA CD 63 28    94  CALL %TPCHK
6DED 4F          95  LD   C,A
6DEE 3E 0F       96  LD   A,BADDEVICE
6DF0 CA 08 73    97  JP   Z,ERROR
6DF3 DD 71 03    98  LD   (IX+DEVICE),C ;**DEVICE
6DF6 CD EB 70    99  CALL SCHDIR
6DF9 DA 08 73   100  JP   C,ERROR
6DFC DD 72 05   101  LD   (IX+DIRPT),D ;**DIRPT
6DFF 1C         102  INC  E
6E00 1D         103  DEC  E
6E01 20 36      104  JR   NZ,FOPEN3
6E03 EB         105  EX   DE,HL
6E04 FD E5      106  PUSH IY
6E06 FD 21 90   107  LD   IY,WORK0
6E09 6D
6E0A 11 0E 00   108  LD   DE,WORK1-WORK0
6E0D 06 04      109  LD   B,4
6E0F            110 FOPEN1:
6E0F FD 7E 01   111  LD   A,(IY+OPENFLG)
6E12 B7         112  OR   A
6E13 28 1D      113  JR   Z,FOPEN2
6E15 FD 7E 03   114  LD   A,(IY+DEVICE)
6E18 DD BE 03   115  CP   (IX+DEVICE)
6E1B 20 15      116  JR   NZ,FOPEN2
6E1D FD 7E 05   117  LD   A,(IY+DIRPT)
6E20 BC         118  CP   H
6E21 20 0F      119  JR   NZ,FOPEN2
6E23 FD 7E 02   120  LD   A,(IY+MODE)
6E26 DD B6 02   121  OR   (IX+MODE)
6E29 28 07      122  JR   Z,FOPEN2
6E2B FD E1      123  POP  IY
6E2D 3E 13      124  LD   A,DOUBLEOPEN
6E2F C3 08 73   125  JP   ERROR
6E32            126 FOPEN2:.
6E32 FD 19      127  ADD  IY,DE
6E34 10 D9      128  DJNZ FOPEN1
6E36 FD E1      129  POP  IY
6E38 EB         130  EX   DE,HL
6E39            131 FOPEN3:
6E39 DD 7E 02   132  LD   A,(IX+MODE)
6E3C FE 03      133  CP   3
6E3E CA C1 6E   134  JP   Z,FOPEN7
6E41 1C         135  INC  E
6E42 1D         136  DEC  E
6E43 4F         137  LD   C,A
6E44 3E 08      138  LD   A,8 ; file not found
6E46 C2 08 73   139  JP   NZ,ERROR
6E49 2A 74 1F   140  LD   HL,(#IBFAD)
6E4C 7E         141  LD   A,(HL)
6E4D E6 07      142  AND  007H
6E4F FE 04      143  CP   004H
6E51 C2 06 73   144  JP   NZ,BADFILEMODE
6E54 11 12 00   145  LD   DE,18
6E57 19         146  ADD  HL,DE
6E58 5E         147  LD   E,(HL)
6E59 23         148  INC  HL
6E5A 56         149  LD   D,(HL)
6E5B 0C         150  INC  C
6E5C 0D         151  DEC  C
6E5D 28 05      152  JR   Z,FOPEN4
6E5F CB 77      153  BIT  6,A
6E61 C2 02 73   154  JP   NZ,WRITEPROTECTED
6E64            155 FOPEN4:
6E64 DD 73 0C   156  LD   (IX+DSIZE),E
6E67 DD 72 0D   157  LD   (IX+DSIZE+1),D ;**DSIZE
6E6A 01 0B 00   158  LD   BC,30-18-1
6E6D 09         159  ADD  HL,BC
6E6E 7E         160  LD   A,(HL)
6E6F DD 77 0A   161  LD   (IX+CHEAD),A ;**CHEAD
6E72 D5         162  PUSH DE
6E73 CD F4 71   163  CALL FATOPEN
6E76 D1         164  POP  DE
6E77 DA 08 73   165  JP   C,ERROR
6E7A 7A         166  LD   A,D
6E7B B3         167  OR   E
6E7C 20 1E      168  JR   NZ,FOPEN5
6E7E CD 84 72   169  CALL INVFAT
6E81 DA 3D 6F   170  JP   C,FOERROR
6E84 D6 7F      171  SUB  07FH
6E86 4F         172  LD   C,A
6E87 78         173  LD   A,B
6E88 FE 10      174  CP   16
6E8A 3E 14      175  LD   A,TOOLONGFILE
6E8C D2 3D 6F   176  JP   NC,FOERROR
6E8F 78         177  LD   A,B
6E90 07         178  RLCA
6E91 07         179  RLCA
6E92 07         180  RLCA
6E93 07         181  RLCA
6E94 81         182  ADD  A,C
6E95 57         183  LD   D,A
6E96 B7         184  OR   A
6E97 3E 14      185  LD   A,TOOLONGFILE
6E99 CA 3D 6F   186  JP   Z,FOERROR
6E9C            187 FOPEN5:
6E9C DD 73 06   188  LD   (IX+SIZE),E
6E9F DD 72 07   189  LD   (IX+SIZE+1),D ;**SIZE
6EA2 AF         190  XOR  A
6EA3 CD 91 71   191  CALL SCRWRDY
6EA6 D4 00 20   192  CALL NC,#DRDSB
6EA9 DA 3D 6F   193  JP   C,FOERROR
6EAC            194 FOPEN6:
6EAC AF         195  XOR  A
6EAD DD 77 04   196  LD   (IX+SCWFLG),A ;**SCWFLG
6EB0 DD 77 08   197  LD   (IX+POINT),A
6EB3 DD 77 09   198  LD   (IX+POINT+1),A ;**POINT
6EB6 DD 77 0B   199  LD   (IX+ACTSC),A ;**ACTSC
6EB9 DD 36 01   200  LD   (IX+OPENFLG),1 ;**OPENFLG
6EBC 01
6EBD 6F         201  LD   L,A
6EBE C3 FF 72   202  JP   NORMAL
6EC1            203 ;
6EC1            204 FOPEN7:
6EC1 1D         205  DEC  E
6EC2 28 32      206  JR   Z,FOPEN9
6EC4 2A 74 1F   207  LD   HL,(#IBFAD)
6EC7 3E 07      208  LD   A,007H
6EC9 A6         209  AND  (HL)
6ECA FE 04      210  CP   004H
6ECC C2 06 73   211  JP   NZ,BADFILEMODE
6ECF CB 76      212  BIT  6,(HL)
6ED1 C2 02 73   213  JP   NZ,WRITEPROTECTED
6ED4 11 1E 00   214  LD   DE,30
6ED7 19         215  ADD  HL,DE
6ED8 5E         216  LD   E,(HL)
6ED9 DD 73 0A   217  LD   (IX+CHEAD),E
6EDC CD F4 71   218  CALL FATOPEN
6EDF DA 08 73   219  JP   C,ERROR
6EE2 CD 84 72   220  CALL INVFAT
6EE5 DA 3D 6F   221  JP   C,FOERROR
6EE8 01 80 00   222  LD   BC,00080H
6EEB DD 7E 0A   223  LD   A,(IX+CHEAD)
```

```
6EEE              224 FOPEN8:
6EEE 60           225  LD   H,B
6EEF 6F           226  LD   L,A
6EF0 19           227  ADD  HL,DE
6EF1 7E           228  LD   A,(HL)
6EF2 70           229  LD   (HL),B
6EF3 B9           230  CP   C
6EF4 38 F8        231  JR   C,FOPEN8
6EF6              232 ;
6EF6              233 FOPEN9:
6EF6 CD F4 71     234  CALL FATOPEN
6EF9 DA 08 73     235  JP   C,ERROR
6EFC CD B6 72     236  CALL ASSIGNC
6EFF 38 3C        237  JR   C,FOERROR
6F01 2A 74 1F     238  LD   HL,(#IBFAD)
6F04 11 12 00     239  LD   DE,18
6F07 19           240  ADD  HL,DE
6F08 36 01        241  LD   (HL),1
6F0A 06 0B        242  LD   B,11
6F0C              243 FOPEN10:
6F0C 23           244  INC  HL
6F0D 72           245  LD   (HL),D
6F0E 10 FC        246  DJNZ FOPEN10
6F10 23           247  INC  HL
6F11 77           248  LD   (HL),A
6F12 DD 77 0A     249  LD   (IX+CHEAD),A ; **CHEAD
6F15 0E 01        250  LD   C,1
6F17 CD 5F 71     251  CALL RWDIR
6F1A 38 21        252  JR   C,FOERROR
6F1C EB           253  EX   DE,HL
6F1D 2A 74 1F     254  LD   HL,(#IBFAD)
6F20 01 20 00     255  LD   BC,32
6F23 ED B0        256  LDIR
6F25 CD 5F 71     257  CALL RWDIR
6F28 38 13        258  JR   C,FOERROR
6F2A DD 36 06     259  LD   (IX+SIZE),1
6F2D 01
6F2E DD 36 07     260  LD   (IX+SIZE+1),0 ;**SIZE
6F31 00
6F32 DD 36 0C     261  LD   (IX+DSIZE),1
6F35 01
6F36 DD 36 0D     262  LD   (IX+DSIZE+1),0 ;**DSIZE
6F39 00
6F3A C3 AC 6E     263  JP   FOPEN6
6F3D              264 ;
6F3D              265 FOERROR:
6F3D F5           266  PUSH AF
6F3E CD 2E 72     267  CALL FATCLOSE
6F41 F1           268  POP  AF
6F42 C3 08 73     269  JP   ERROR
6F45              270 ;
6F45              271 ;
6F45              272 FSEEK:
6F45 CD DC 72     273  CALL SETIX
6F48 DA 91 6F     274  JP   C,FSERROR
6F4B 3E 0C        275  LD   A,12 ; file not open
6F4D CA 91 6F     276  JP   Z,FSERROR
6F50 3E 0E        277  LD   A,14 ; bad data
6F52 04           278  INC  B
6F53 05           279  DEC  B
6F54 C2 91 6F     280  JP   NZ,FSERROR
6F57 0C           281  INC  C
6F58 0D           282  DEC  C
6F59 28 20        283  JR   Z,FSEEK3
6F5B 0D           284  DEC  C
6F5C 28 0D        285  JR   Z,FSEEK1
6F5E 0D           286  DEC  C
6F5F 20 30        287  JR   NZ,FSERROR
6F61              288 ;
6F61 DD 6E 06     289  LD   L,(IX+SIZE)
6F64 DD 66 07     290  LD   H,(IX+SIZE+1)
6F67 19           291  ADD  HL,DE
6F68 EB           292  EX   DE,HL
6F69 18 10        293  JR   FSEEK3
6F6B              294 ;
6F6B              295 FSEEK1:
6F6B DD 6E 08     296  LD   L,(IX+POINT)
6F6E DD 66 09     297  LD   H,(IX+POINT+1)
6F71 19           298  ADD  HL,DE
6F72 CB 7A        299  BIT  7,D
6F74 28 01        300  JR   Z,FSEEK2
6F76 3F           301  CCF
6F77              302 FSEEK2:
6F77 DA 8F 6F     303  JP   C,FSOUTOFFILE
6F7A EB           304  EX   DE,HL
6F7B              305 ;
6F7B              306 FSEEK3:
6F7B DD 6E 06     307  LD   L,(IX+SIZE)
6F7E DD 66 07     308  LD   H,(IX+SIZE+1)
6F81 AF           309  XOR  A
6F82 ED 52        310  SBC  HL,DE
6F84 DA 8F 6F     311  JP   C,FSOUTOFFILE
6F87 DD 73 08     312  LD   (IX+POINT),E
6F8A DD 72 09     313  LD   (IX+POINT+1),D
6F8D EB           314  EX   DE,HL
6F8E C9           315  RET
6F8F              316 ;
6F8F              317 FSOUTOFFILE:
6F8F 3E 11        318  LD   A,OUTOFFILE
6F91              319 FSERROR:
6F91 21 FF FF     320  LD   HL,0FFFFH
6F94 37           321  SCF
6F95 C9           322  RET
6F96              323 ;
6F96              324 ;
6F96              325 FPUTC:
6F96 0E 00        326  LD   C,0
6F98 CD B0 6F     327  CALL FPG
6F9B 73           328  LD   (HL),E
6F9C DD 36 04     329  LD   (IX+SCWFLG),1
6F9F 01
6FA0 2E 00        330  LD   L,0
6FA2 C3 FF 72     331  JP   NORMAL
6FA5              332 ;
6FA5              333 ;
6FA5              334 FGETC:
6FA5 16 00        335  LD   D,0
6FA7 0E 01        336  LD   C,1
6FA9 CD B0 6F     337  CALL FPG
6FAC 6E           338  LD   L,(HL)
6FAD C3 FF 72     339  JP   NORMAL
6FB0              340 ;
6FB0              341 ;
6FB0              342 FPG:
```

```
6FB0 D5           343  PUSH DE
6FB1 CD DC 72     344  CALL SETIX
6FB4 DA 8A 70     345  JP   C,FPGERROR
6FB7 20 05        346  JR   NZ,FPG1
6FB9 3E 0C        347  LD   A,12 ; file not open
6FBB C3 8A 70     348  JP   FPGERROR
6FBE              349 FPG1:
6FBE DD 7E 02     350  LD   A,(IX+MODE)
6FC1 B9           351  CP   C
6FC2 3E 10        352  LD   A,OPENMODEERROR
6FC4 CA 8A 70     353  JP   Z,FPGERROR
6FC7 14           354  INC  D
6FC8 15           355  DEC  D
6FC9 28 05        356  JR   Z,FPG2
6FCB 3E 0E        357  LD   A,14 ; bad data
6FCD C3 8A 70     358  JP   FPGERROR
6FD0              359 ;
6FD0              360 FPG2:
6FD0 DD 5E 08     361  LD   E,(IX+POINT)
6FD3 DD 7E 06     362  LD   A,(IX+SIZE)
6FD6 BB           363  CP   E
6FD7 20 45        364  JR   NZ,FPG5
6FD9 DD 56 09     365  LD   D,(IX+POINT+1)
6FDC DD 7E 07     366  LD   A,(IX+SIZE+1)
6FDF BA           367  CP   D
6FE0 20 3C        368  JR   NZ,FPG5
6FE2 0D           369  DEC  C
6FE3 3E 11        370  LD   A,OUTOFFILE
6FE5 CA 8A 70     371  JP   Z,FPGERROR
6FE8 7A           372  LD   A,D
6FE9 A3           373  AND  E
6FEA 3C           374  INC  A
6FEB 3E 14        375  LD   A,TOOLONGFILE
6FED CA 8A 70     376  JP   Z,FPGERROR
6FF0 1C           377  INC  E
6FF1 1D           378  DEC  E
6FF2 20 23        379  JR   NZ,FPG4
6FF4 CD A6 72     380  CALL GETFATNO
6FF7 7A           381  LD   A,D
6FF8 E6 0F        382  AND  0FH
6FFA CB FF        383  SET  7,A
6FFC 20 06        384  JR   NZ,FPG3
6FFE CD B6 72     385  CALL ASSIGNC
7001 DA 8A 70     386  JP   C,FPGERROR
7004              387 FPG3:
7004 08           388  EX   AF,AF'
7005 D5           389  PUSH DE
7006 CD 84 72     390  CALL INVFAT
7009 D1           391  POP  DE
700A DA 8A 70     392  JP   C,FPGERROR
700D 08           393  EX   AF,AF'
700E 77           394  LD   (HL),A
700F 21 89 6D     395  LD   HL,FBWRFLG
7012 06 00        396  LD   B,0
7014 09           397  ADD  HL,BC
7015 36 01        398  LD   (HL),1
7017              399 FPG4:
7017 13           400  INC  DE
7018 DD 73 06     401  LD   (IX+SIZE),E
701B DD 72 07     402  LD   (IX+SIZE+1),D
701E              403 ;
701E              404 FPG5:
701E 3A 80 6D     405  LD   A,(CMODE)
7021 B7           406  OR   A
7022 DD 7E 09     407  LD   A,(IX+POINT+1)
7025 28 02        408  JR   Z,FPG6
7027 E6 F0        409  AND  0F0H
7029              410 FPG6:
7029 DD BE 0B     411  CP   (IX+ACTSC)
702C 28 2E        412  JR   Z,FPG9
702E DD 7E 04     413  LD   A,(IX+SCWFLG)
7031 B7           414  OR   A
7032 28 10        415  JR   Z,FPG7
7034 DD 7E 0B     416  LD   A,(IX+ACTSC)
7037 CD 91 71     417  CALL SCRWRDY
703A D4 03 20     418  CALL NC,#DWTSB
703D DA 8A 70     419  JP   C,FPGERROR
7040 DD 36 04     420  LD   (IX+SCWFLG),0
7043 00
7044              421 FPG7:
7044 DD 7E 09     422  LD   A,(IX+POINT+1)
7047 CD 91 71     423  CALL SCRWRDY
704A 4F           424  LD   C,A
704B D4 00 20     425  CALL NC,#DRDSB
704E DA 8A 70     426  JP   C,FPGERROR
7051 DD 7E 09     427  LD   A,(IX+POINT+1)
7054 0D           428  DEC  C
7055 28 02        429  JR   Z,FPG8
7057 E6 F0        430  AND  0F0H
7059              431 FPG8:
7059 DD 77 0B     432  LD   (IX+ACTSC),A
705C              433 ;
705C              434 FPG9:
705C 2A 83 6D     435  LD   HL,(SCBUFAD)
705F 3A 80 6D     436  LD   A,(CMODE)
7062 B7           437  OR   A
7063 DD 7E 00     438  LD   A,(IX+FNO)
7066 28 0B        439  JR   Z,FPG10
7068 87           440  ADD  A,A
7069 87           441  ADD  A,A
706A 87           442  ADD  A,A
706B 87           443  ADD  A,A
706C 5F           444  LD   E,A
706D DD 7E 09     445  LD   A,(IX+POINT+1)
7070 E6 0F        446  AND  00FH
7072 83           447  ADD  A,E
7073              448 FPG10:
7073 84           449  ADD  A,H
7074 67           450  LD   H,A
7075 DD 5E 08     451  LD   E,(IX+POINT)
7078 16 00        452  LD   D,0
707A 19           453  ADD  HL,DE
707B DD 5E 08     454  LD   E,(IX+POINT)
707E DD 56 09     455  LD   D,(IX+POINT+1)
7081 13           456  INC  DE
7082 DD 73 08     457  LD   (IX+POINT),E
7085 DD 72 09     458  LD   (IX+POINT+1),D
7088 D1           459  POP  DE
7089 C9           460  RET
708A              461 ;
708A              462 FPGERROR:
708A E1           463  POP  HL
708B E1           464  POP  HL
708C C3 08 73     465  JP   ERROR
```

▶ 大学のサークル（オーケストラ）でOh!Xを読んでいたら皆の見る目が変わってしまった。
後藤　和正(19)埼玉県

```
708F                 466 ;
708F                 467 ;
708F                 468 FCLOSE:
708F CD DC 72        469  CALL SETIX
7092 DA 08 73        470  JP   C,ERROR
7095 3E 0C           471  LD   A,12 ; file not open
7097 CA 08 73        472  JP   Z,ERROR
709A DD 7E 04        473  LD   A,(IX+SCWFLG)
709D B7              474  OR   A
709E 28 0C           475  JR   Z,FCLOSE1
70A0 DD 7E 0B        476  LD   A,(IX+ACTSC)
70A3 CD 91 71        477  CALL SCRWRDY
70A6 D4 03 20        478  CALL NC,#DWTSB
70A9 DA 08 73        479  JP   C,ERROR
70AC                 480 FCLOSE1:
70AC DD 7E 06        481  LD   A,(IX+SIZE)
70AF DD BE 0C        482  CP   (IX+DSIZE)
70B2 20 08           483  JR   NZ,FCLOSE2
70B4 DD 7E 07        484  LD   A,(IX+SIZE+1)
70B7 DD BE 0D        485  CP   (IX+DSIZE+1)
70BA 28 1C           486  JR   Z,FCLOSE3
70BC                 487 FCLOSE2:
70BC 0E 01           488  LD   C,1
70BE CD 5F 71        489  CALL RWDIR
70C1 DA 08 73        490  JP   C,ERROR
70C4 11 12 00        491  LD   DE,18
70C7 19              492  ADD  HL,DE
70C8 DD 7E 06        493  LD   A,(IX+SIZE)
70CB 77              494  LD   (HL),A
70CC DD 7E 07        495  LD   A,(IX+SIZE+1)
70CF 23              496  INC  HL
70D0 77              497  LD   (HL),A
70D1 4A              498  LD   C,D
70D2 CD 5F 71        499  CALL RWDIR
70D5 DA 08 73        500  JP   C,ERROR
70D8                 501 FCLOSE3:
70D8 DD 36 01        502  LD   (IX+OPENFLG),0
70DB 00
70DC CD 2E 72        503  CALL FATCLOSE
70DF 2E 00           504  LD   L,0
70E1 D2 FF 72        505  JP   NC,NORMAL
70E4 DD 36 01        506  LD   (IX+OPENFLG),1
70E7 01
70E8 C3 08 73        507  JP   ERROR
70EB                 508 ;
70EB                 509 ; SCHDIR
70EB                 510 ;   in --- (#IBFAD), #DSK
70EB                 511 ;   out -- D = directory offset
70EB                 512 ;          E = found.......0
70EB                 513 ;              not found...1
70EB                 514 ;          Cy = error flag
70EB                 515 ;   break -- AF, BC, HL, AF'
70EB                 516 ;
70EB                 517 SCHDIR:
70EB 3E FF           518  LD   A,0FFH
70ED 32 8D 6D        519  LD   (SPACEOS),A
70F0 0E 00           520  LD   C,0
70F2 ED 5B 60        521  LD   DE,(#DIRPS)
70F5 1F
70F6 21 10 00        522  LD   HL,16
70F9 19              523  ADD  HL,DE
70FA 22 8E 6D        524  LD   (DIREND),HL
70FD                 525 ;
70FD                 526 SCHDIR1:
70FD 2A 64 1F        527  LD   HL,(#DTBUF)
7100 3E 01           528  LD   A,1
7102 CD 00 20        529  CALL #DRDSB
7105 D8              530  RET  C
7106 06 08           531  LD   B,8
7108                 532 ;
7108                 533 SCHDIR2:
7108 7E              534  LD   A,(HL)
7109 3C              535  INC  A
710A 28 46           536  JR   Z,SCHDIR8
710C 3D              537  DEC  A
710D 20 0C           538  JR   NZ,SCHDIR3
710F 3A 8D 6D        539  LD   A,(SPACEOS)
7112 3C              540  INC  A
7113 20 2A           541  JR   NZ,SCHDIR6
7115 79              542  LD   A,C
7116 32 8D 6D        543  LD   (SPACEOS),A
7119 18 24           544  JR   SCHDIR6
711B                 545 SCHDIR3:
711B E5              546  PUSH HL
711C D5              547  PUSH DE
711D C5              548  PUSH BC
711E ED 5B 74        549  LD   DE,(#IBFAD)
7121 1F
7122 06 10           550  LD   B,16
7124                 551 SCHDIR4:
7124 23              552  INC  HL
7125 13              553  INC  DE
7126 1A              554  LD   A,(DE)
7127 BE              555  CP   (HL)
7128 20 02           556  JR   NZ,SCHDIR5
712A 10 F8           557  DJNZ SCHDIR4
712C                 558 SCHDIR5:
712C C1              559  POP  BC
712D D1              560  POP  DE
712E E1              561  POP  HL
712F 20 0E           562  JR   NZ,SCHDIR6
7131 79              563  LD   A,C
7132 ED 5B 74        564  LD   DE,(#IBFAD)
7135 1F
7136 01 20 00        565  LD   BC,32
7139 ED B0           566  LDIR
713B 57              567  LD   D,A
713C 58              568  LD   E,B
713D B7              569  OR   A
713E C9              570  RET     ; found
713F                 571 ;
713F                 572 SCHDIR6:
713F 0C              573  INC  C
7140 3E 20           574  LD   A,32
7142 85              575  ADD  A,L
7143 6F              576  LD   L,A
7144 30 01           577  JR   NC,SCHDIR7
7146 24              578  INC  H
7147                 579 SCHDIR7:
7147 10 BF           580  DJNZ SCHDIR2
7149                 581 ;
7149 13              582  INC  DE
714A 2A 8E 6D        583  LD   HL,(DIREND)
714D B7              584  OR   A
714E ED 52           585  SBC  HL,DE
7150 20 AB           586  JR   NZ,SCHDIR1
7152                 587 ;
7152                 588 SCHDIR8:
7152 3A 8D 6D        589  LD   A,(SPACEOS)
7155 FE FF           590  CP   0FFH
7157 20 01           591  JR   NZ,SCHDIR9
7159 79              592  LD   A,C
715A                 593 SCHDIR9:
715A 57              594  LD   D,A
715B 1E 01           595  LD   E,1
715D B7              596  OR   A
715E C9              597  RET     ; not found
715F                 598 ;
715F                 599 ; RWDIR
715F                 600 ;   if C=0 write else read and let HL=IB adr
715F                 601 ;
715F                 602 RWDIR:
715F DD 7E 03        603  LD   A,(IX+DEVICE)
7162 32 5D 1F        604  LD   (#DSK),A
7165 DD 7E 05        605  LD   A,(IX+DIRPT)
7168 0F              606  RRCA
7169 0F              607  RRCA
716A 0F              608  RRCA
716B 47              609  LD   B,A
716C E6 1F           610  AND  01FH
716E ED 5B 60        611  LD   DE,(#DIRPS)
7171 1F
7172 83              612  ADD  A,E
7173 5F              613  LD   E,A
7174 30 01           614  JR   NC,RWDIR1
7176 14              615  INC  D
7177                 616 RWDIR1:
7177 2A 64 1F        617  LD   HL,(#DTBUF)
717A 3E 01           618  LD   A,1
717C 0C              619  INC  C
717D 0D              620  DEC  C
717E 20 04           621  JR   NZ,RWDIR2
7180 CD 03 20        622  CALL #DWTSB
7183 C9              623  RET
7184                 624 RWDIR2:
7184 CD 00 20        625  CALL #DRDSB
7187 D8              626  RET  C
7188 78              627  LD   A,B
7189 E6 E0           628  AND  0E0H
718B 85              629  ADD  A,L
718C 6F              630  LD   L,A
718D D0              631  RET  NC
718E 24              632  INC  H
718F B7              633  OR   A
7190 C9              634  RET
7191                 635 ;
7191                 636 ; SCRWRDY
7191                 637 ;   in --- A = high address
7191                 638 ;   out -- A = records
7191                 639 ;          DE = record No.
7191                 640 ;          HL = (SCBUF(file No.))
7191                 641 ;
7191                 642 SCRWRDY:
7191 DD 5E 0A        643  LD   E,(IX+CHEAD)
7194 57              644  LD   D,A
7195 E6 F0           645  AND  0F0H
7197 28 28           646  JR   Z,SCRWRDY4
7199 0F              647  RRCA
719A 0F              648  RRCA
719B 0F              649  RRCA
719C 0F              650  RRCA
719D F5              651  PUSH AF
719E CD A6 72        652  CALL GETFATNO
71A1 79              653  LD   A,C
71A2 CD D6 72        654  CALL GETFBAD
71A5 C1              655  POP  BC
71A6 4A              656  LD   C,D
71A7 16 00           657  LD   D,0
71A9 EB              658  EX   DE,HL
71AA                 659 SCRWRDY1:
71AA 19              660  ADD  HL,DE
71AB 7E              661  LD   A,(HL)
71AC B7              662  OR   A
71AD 28 04           663  JR   Z,SCRWRDY2
71AF FE 80           664  CP   080H
71B1 38 04           665  JR   C,SCRWRDY3
71B3                 666 SCRWRDY2:
71B3 3E 07           667  LD   A,7 ; bad allocation table
71B5 37              668  SCF
71B6 C9              669  RET
71B7                 670 SCRWRDY3:
71B7 26 00           671  LD   H,0
71B9 6F              672  LD   L,A
71BA 10 EE           673  DJNZ SCRWRDY1
71BC 79              674  LD   A,C
71BD E6 0F           675  AND  00FH
71BF 57              676  LD   D,A
71C0 5D              677  LD   E,L
71C1                 678 SCRWRDY4:
71C1 DD 7E 03        679  LD   A,(IX+DEVICE)
71C4 32 5D 1F        680  LD   (#DSK),A
71C7 4A              681  LD   C,D
71C8 7B              682  LD   A,E
71C9 07              683  RLCA
71CA 07              684  RLCA
71CB 07              685  RLCA
71CC 07              686  RLCA
71CD 5F              687  LD   E,A
71CE E6 0F           688  AND  00FH
71D0 57              689  LD   D,A
71D1 7B              690  LD   A,E
71D2 E6 F0           691  AND  0F0H
71D4 81              692  ADD  A,C
71D5 5F              693  LD   E,A
71D6 2A 83 6D        694  LD   HL,(SCBUFAD)
71D9 3A 80 6D        695  LD   A,(CMODE)
71DC B7              696  OR   A
71DD DD 7E 00        697  LD   A,(IX+FNO)
71E0 20 05           698  JR   NZ,SCRWRDY5
71E2 84              699  ADD  A,H
71E3 67              700  LD   H,A
71E4 AF              701  XOR  A
71E5 3C              702  INC  A
71E6 C9              703  RET
71E7                 704 SCRWRDY5:
71E7 87              705  ADD  A,A
71E8 87              706  ADD  A,A
71E9 87              707  ADD  A,A
```

▶う～ん，3D 2（仮）は期待大ですね。　　佐藤　敏幸(26)神奈川県

```
71EA 87          708  ADD  A,A
71EB 84          709  ADD  A,H
71EC 67          710  LD   H,A
71ED 7B          711  LD   A,E
71EE E6 F0       712  AND  0F0H
71F0 5F          713  LD   E,A
71F1 3E 10       714  LD   A,16
71F3 C9          715  RET
71F4             716 ;
71F4             717 ; FATOPEN
71F4             718 ;    out -- C = FAT No.
71F4             719 ;
71F4             720 FATOPEN:
71F4 CD A6 72    721  CALL GETFATNO
71F7 79          722  LD   A,C
71F8 FE 04       723  CP   4
71FA 3F          724  CCF
71FB D0          725  RET  NC
71FC 21 84 6D    726  LD   HL,FBUFDEV-1
71FF 0E FF       727  LD   C,-1
7201 AF          728  XOR  A
7202             729 FATOPEN1:
7202 23          730  INC  HL
7203 0C          731  INC  C
7204 BE          732  CP   (HL)
7205 20 FB       733  JR   NZ,FATOPEN1
7207 DD 7E 03    734  LD   A,(IX+DEVICE)
720A 32 5D 1F    735  LD   (#DSK),A
720D 47          736  LD   B,A
720E 79          737  LD   A,C
720F CD D6 72    738  CALL GETFBAD
7212 ED 5B 5E    739  LD   DE,(#FATPOS)
7215 1F
7216 3E 01       740  LD   A,1
7218 CD 00 20    741  CALL #DRDSB
721B D8          742  RET  C
721C 21 85 6D    743  LD   HL,FBUFDEV
721F 79          744  LD   A,C
7220 85          745  ADD  A,L
7221 6F          746  LD   L,A
7222 30 01       747  JR   NC,FATOPEN2
7224 24          748  INC  H
7225             749 FATOPEN2:
7225 70          750  LD   (HL),B
7226 11 04 00    751  LD   DE,FBWRFLG-FBUFDEV
7229 19          752  ADD  HL,DE
722A 36 00       753  LD   (HL),0
722C B7          754  OR   A
722D C9          755  RET
722E             756 ;
722E             757 ;
722E             758 FATCLOSE:
722E DD 4E 03    759  LD   C,(IX+DEVICE)
7231 DD E5       760  PUSH IX
7233 DD 21 90    761  LD   IX,WORK0
7236 6D
7237 11 0E 00    762  LD   DE,WORK1-WORK0
723A 2E 01       763  LD   L,1
723C 06 04       764  LD   B,4
723E             765 FATCLOSE1:
723E DD 7E 01    766  LD   A,(IX+OPENFLG)
7241 B7          767  OR   A
7242 28 07       768  JR   Z,FATCLOSE2
7244 DD 7E 03    769  LD   A,(IX+DEVICE)
7247 B9          770  CP   C
7248 20 01       771  JR   NZ,FATCLOSE2
724A 2C          772  INC  L
724B             773 FATCLOSE2:
724B DD 19       774  ADD  IX,DE
724D 10 EF       775  DJNZ FATCLOSE1
724F DD E1       776  POP  IX
7251 B7          777  OR   A
7252 2D          778  DEC  L
7253 C0          779  RET  NZ
7254             780 ;
7254 CD A6 72    781  CALL GETFATNO
7257 21 89 6D    782  LD   HL,FBWRFLG
725A 79          783  LD   A,C
725B 85          784  ADD  A,L
725C 6F          785  LD   L,A
725D 30 01       786  JR   NC,FATCLOSE3
725F 24          787  INC  H
7260             788 FATCLOSE3:
7260 7E          789  LD   A,(HL)
7261 11 04 00    790  LD   DE,FBWRFLG-FBUFDEV
7264 B7          791  OR   A
7265 ED 52       792  SBC  HL,DE
7267 36 00       793  LD   (HL),0
7269 B7          794  OR   A
726A C8          795  RET  Z
726B             796 ;
726B E5          797  PUSH HL
726C DD 7E 03    798  LD   A,(IX+DEVICE)
726F 32 5D 1F    799  LD   (#DSK),A
7272 47          800  LD   B,A
7273 79          801  LD   A,C
7274 CD D6 72    802  CALL GETFBAD
7277 ED 5B 5E    803  LD   DE,(#FATPOS)
727A 1F
727B 3E 01       804  LD   A,1
727D CD 03 20    805  CALL #DWTSB
7280 E1          806  POP  HL
7281 D0          807  RET  NC
7282 70          808  LD   (HL),B
7283 C9          809  RET
7284             810 ;
7284             811 ; INVFAT
7284             812 ;    in --- C = FAT No.
7284             813 ;    out -- A = last data
7284             814 ;            B = clusters-1
7284             815 ;            DE = SFBUF address
7284             816 ;            HL = address of last data
7284             817 ;    break -- F
7284             818 ;
7284             819 INVFAT:
7284 79          820  LD   A,C
7285 CD D6 72    821  CALL GETFBAD
7288 EB          822  EX   DE,HL
7289 26 00       823  LD   H,0
728B DD 6E 0A    824  LD   L,(IX+CHEAD)
728E 44          825  LD   B,H
```

```
728F             826 INVFAT1:
728F 19          827  ADD  HL,DE
7290 7E          828  LD   A,(HL)
7291 B7          829  OR   A
7292 28 0E       830  JR   Z,INVFAT3
7294 FE 80       831  CP   080H
7296 30 06       832  JR   NC,INVFAT2
7298 04          833  INC  B
7299 6F          834  LD   L,A
729A 26 00       835  LD   H,0
729C 18 F1       836  JR   INVFAT1
729E             837 INVFAT2:
729E FE 90       838  CP   090H
72A0 3F          839  CCF
72A1 D0          840  RET  NC
72A2             841 INVFAT3:
72A2 3E 07       842  LD   A,7 ; bad allocation table
72A4 37          843  SCF
72A5 C9          844  RET
72A6             845 ;
72A6             846 ;
72A6             847 GETFATNO:
72A6 21 85 6D    848  LD   HL,FBUFDEV
72A9 01 00 04    849  LD   BC,4*256+0
72AC DD 7E 03    850  LD   A,(IX+DEVICE)
72AF             851 GETFATNO1:
72AF BE          852  CP   (HL)
72B0 C8          853  RET  Z
72B1 23          854  INC  HL
72B2 0C          855  INC  C
72B3 10 FA       856  DJNZ GETFATNO1
72B5 C9          857  RET
72B6             858 ;
72B6             859 ; ASSIGNC
72B6             860 ;    in --- C = FAT No.
72B6             861 ;    out -- A = cluster
72B6             862 ;    break -- F, B, HL
72B6             863 ;
72B6             864 ASSIGNC:
72B6 79          865  LD   A,C
72B7 CD D6 72    866  CALL GETFBAD
72BA AF          867  XOR  A
72BB 06 80       868  LD   B,128
72BD             869 ASSIGNC1:
72BD BE          870  CP   (HL)
72BE 28 07       871  JR   Z,ASSIGNC2
72C0 23          872  INC  HL
72C1 10 FA       873  DJNZ ASSIGNC1
72C3 3E 09       874  LD   A,9 ; device full
72C5 37          875  SCF
72C6 C9          876  RET
72C7             877 ASSIGNC2:
72C7 36 80       878  LD   (HL),080H
72C9 3E 80       879  LD   A,128
72CB 90          880  SUB  B
72CC 06 00       881  LD   B,0
72CE 21 89 6D    882  LD   HL,FBWRFLG
72D1 09          883  ADD  HL,BC
72D2 36 01       884  LD   (HL),1
72D4 B7          885  OR   A
72D5 C9          886  RET
72D6             887 ;
72D6             888 ; GETFBAD
72D6             889 ;    in --- A = FAT No.
72D6             890 ;    out -- HL = FBUF address
72D6             891 ;    break -- AF
72D6             892 ;
72D6             893 GETFBAD:
72D6 2A 81 6D    894  LD   HL,(FBUFAD)
72D9 84          895  ADD  A,H
72DA 67          896  LD   H,A
72DB C9          897  RET
72DC             898 ;
72DC             899 ;
72DC             900 SETIX:
72DC 7D          901  LD   A,L
72DD E6 FC       902  AND  0FCH
72DF B4          903  OR   H
72E0 3E 0E       904  LD   A,14 ; bad data
72E2 37          905  SCF
72E3 C0          906  RET  NZ
72E4             907 ;
72E4 D5          908  PUSH DE
72E5 11 0E 00    909  LD   DE,WORK1-WORK0
72E8 DD 21 90    910  LD   IX,WORK0
72EB 6D
72EC 7D          911  LD   A,L
72ED 0F          912  RRCA
72EE 30 02       913  JR   NC,SETIX1
72F0 DD 19       914  ADD  IX,DE
72F2             915 SETIX1:
72F2 0F          916  RRCA
72F3 30 04       917  JR   NC,SETIX2
72F5 DD 19       918  ADD  IX,DE
72F7 DD 19       919  ADD  IX,DE
72F9             920 SETIX2:
72F9 D1          921  POP  DE
72FA DD 7E 01    922  LD   A,(IX+OPENFLG)
72FD B7          923  OR   A
72FE C9          924  RET
72FF             925 ;
72FF             926 ;
72FF             927 NORMAL:
72FF AF          928  XOR  A
7300 67          929  LD   H,A
7301 C9          930  RET
7302             931 ;
7302             932 ;
7302             933 WRITEPROTECTED:
7302 3E 04       934  LD   A,4
7304 18 02       935  JR   ERROR
7306             936 ;
7306             937 BADFILEMODE:
7306 3E 06       938  LD   A,6
7308             939 ;
7308             940 ERROR:
7308 26 FF       941  LD   H,0FFH
730A 6F          942  LD   L,A
730B 37          943  SCF
730C C9          944  RET
```

全機種共通 システムインデックス

■85年6月号
序論　共通化の試み
第1部　S-OS"MACE"
第2部　Lisp-85インタプリタ
第3部　チェックサムプログラム
■85年7月号
第4部　マシン語プログラム開発入門
第5部　エディタアセンブラZEDA
第6部　デバッグツールZAID
■85年8月号
第7部　ゲーム開発パッケージBEMS
第8部　ソースジェネレータZING
■85年9月号
インタラプト　S-OS番外地
第9部　マシン語入力ツールMACINTO-S
第10部　Lisp-85入門(1)
■85年10月号
第11部　仮想マシンCAP-X85
連載　Lisp-85入門(2)
■85年11月号
連載　Lisp-85入門(3)
■85年12月号
第12部　Prolog-85発表
■86年1月号
第13部　リロケータブルのお話
第14部　FM音源サウンドエディタ
■86年2月号
第15部　S-OS"SWORD"
第16部　Prolog-85入門(1)
■86年3月号
第17部　magiFORTH発表
連載　Prolog-85入門(2)
■86年4月号
第18部　思考ゲームJEWEL
第19部　LIFE GAME
連載　基礎からのmagiFORTH
連載　Prolog-85入門(3)
■86年5月号
第20部　スクリーンエディタE-MATE
連載　実戦演習magiFORTH
■86年6月号
第21部　Z80TRACER
第22部　magiFORTH TRACER
第23部　ディスクダンプ&エディタ
第24部　"SWORD" 2000 QD
連載　対話で学ぶ magiFORTH
特別付録　PC-8801版S-OS"SWORD"
■86年7月号
第25部　FM音源ミュージックシステム
付録　FM音源ボードの製作
連載　計算力アップのmagiFORTH
特別付録　SMC-777版 S-OS"SWORD"
■86年8月号
第26部　対局五目並べ
第27部　MZ-2500版 S-OS"SWORD"
■86年9月号
第28部　FuzzyBASIC 発表
連載　明日に向かって magiFORTH
■86年10月号
第29部　ちょっと便利な拡張プログラム
第30部　ディスクモニタ DREAM
第31部　FuzzyBASIC 料理法<1>
■86年11月号
第32部　パズルゲーム HOTTAN
第33部　MAZE in MAZE
連載　FuzzyBASIC 料理法<2>
■86年12月号
第34部　CASL & COMET
連載　FuzzyBASIC 料理法<3>
■87年1月号
第35部　マシン語入力ツールMACINTO-C
連載　FuzzyBASIC 料理法<4>
■87年2月号
第36部　アドベンチャーゲーム MARMALADE
第37部　テキアペ作成ツール CONTEX
■87年3月号
第38部　魔法使いはアニメがお好き
第39部　アニメーションツール MAGE
付録　"SWORD" 再掲載と MAGIC の標準化
■87年4月号
第40部　INVADER GAME
第41部　TANGERINE
■87年5月号
第42部　S-OS"SWORD" 変身セット
第43部　MZ-700用 "SWORD" を QD 対応に
■87年6月号
インタラプト　コンパイラ物語
第44部　FuzzyBASIC コンパイラ
第45部　エディタアセンブラ ZEDA-3
■87年7月号
第46部　STORY MASTER
■87年8月号
第47部　パズルゲーム碁石拾い
第48部　漢字出力パッケージ JACKWRITE
特別付録　FM-7/77版 S-OS"SWORD"
■87年9月号
第49部　リロケータブル逆アセンブラ Inside-R
特別付録　PC-8001/8801 版 S-OS"SWORD"
■87年10月号
第50部　tiny CORE WARS
第51部　FuzzyBASIC コンパイラの拡張
第52部　X1turbo 版 S-OS"SWORD"
■87年11月号
序論　神話のなかのマイクロコンピュータ
付録　S-OS の仲間たち
第53部　もうひとつの FuzzyBASIC 入門
第54部　ファイルアロケータ&ローダ
インタラプト　S-OS こちら集中治療室
第55部　BACK GAMMON
■87年12月号
第56部　タートルグラフィックパッケージTURTLE
第57部　X1turbo 版 "SWORD" アフターケア
ラインプリントルーチン
特別付録　PASOPIA7 版 S-OS"SWORD"
■88年1月号
第58部　FuzzyBASIC コンパイラ・奥村版
付録　石上版コンパイラ拡張部の修正
■88年2月号
第59部　シューティングゲーム ELFES
■88年3月号
第60部　構造型コンパイラ言語 SLANG
■88年4月号
第61部　デバッギングツール TRADE
第62部　シミュレーションウォーゲーム WALRUS
■88年5月号
第63部　シューティングゲーム ELFES II
第64部　地底最大の作戦
■88年6月号
第65部　構造化言語 SLANG 入門(1)
第66部　Lisp-85 用 NAMPA シミュレーション
■88年7月号
第67部　マルチウィンドウドライバ MW-1
連載　構造化言語 SLANG 入門(2)
■88年8月号
第68部　マルチウィンドウエディタ WINER
■88年9月号
第69部　超小型エディタ TED-750
第70部　アフターケア WINER の拡張
■88年10月号
第71部　SLANG 用ファイル入出力ライブラリ
第72部　シューティングゲーム MANKAI
■88年11月号
第73部　シューティングゲーム ELFES Ⅳ
■88年12月号
第74部　ソースジェネレータ SOURCERY
■89年1月号
第75部　パズルゲーム LAST ONE
第76部　ブロックゲーム FLICK
■89年2月号
第77部　高速エディタアセンブラ REDA
特別付録　X1版 S-OS"SWORD"<再掲載>
■89年3月号
第78部　Z80用浮動小数点演算パッケージSOROBAN
■89年4月号
第79部　SLANG 用実数演算ライブラリ
■89年5月号
第80部　ソースジェネレータ RING
■89年6月号
第81部　超小型コンパイラTTC
■89年7月号
第82部　TTC用パズルゲーム　TICBAN
■89年8月号
第83部　CP/M用ファイルコンバータ
■89年9月号
第84部　生物進化シミュレーションBUGS
■89年10月号
第85部　小型インタプリタ言語TTI
■89年11月号
第86部　TTI用パズルゲーム　PUSH BON!
■89年12月号
第87部　SLANG用リダイレクションライブラリ
DIO. LIB
■90年1月号
第88部　SLANG用ゲームWORM KUN
特別付録　再掲載SLANGコンパイラ
■90年2月号
第89部　超小型コンパイラTTC++
■90年3月号
第90部　超多機能アセンブラOHM-Z80
■90年4月号
第91部　ファジィコンピュータシミュレーションI-MY
■90年5月号
第92部　インタプリタ言語STACK
■90年6月号
第93部　リロケータブルフォーマットの取り決め
第94部　STACK用ゲーム　SQUASH！
第95部　X68000対応S-OS"SWORD"
特別付録　PC-286対応S-OS"SWORD"
■90年7月号
第96部　リロケータブルアセンブラWZD
■90年8月号
第97部　リンカWLK
■90年9月号
第98部　BILLIARDS
■90年10月号
第99部　ライブラリアンWLB
■90年11月号
第100部　タブコード対応エディタEDC-T
■90年12月号
第101部　STACKコンパイラ
■91年1月号
第102部　ブロックアクションゲーム COLUMNS
■91年2月号
第103部　ダイスゲームKISMET
■91年3月号
第104部　アクションゲームMUD BALLIN'
■91年4月号
第105部　SLANG用カードゲームDOBON
■91年5月号
第106部　実数型コンパイラ言語REAL
■91年6月号
第107部　Small-C処理系の移植
■91年7月号
第108部　REAL ソースリスト編
■91年8月号
第109部　Small-Cライブラリの移植

INDEX

＊以上のアプリケーションは，基本システムである S-OS "MACE" または S-OS"SWORD" がないと動作しませんのでご注意ください。

以前はパソコンなどコンピュータの取材を頻繁にやっていたのだが，仕事の部署が変わってからは，疎遠になっている。なんたって，今はレジャー，メディア，小売り，飲食，交通といったサービス業について，あれこれと取材しているので，さすがにパソコンにうとい。

ちょっと仕事のパターンを紹介すると，映画の試写を見たあと，カード会社の取材をして，そのあと，コンビニエンスストアに関する調査をして，レーザーディスクの売れ行きをチェック。翌日はバルセロナ五輪についての取材メモをひっくり返してから，成田空港の見学に行く，という始末である。おっと，その間にハンバーガーの試食をしたっけか。とにかくそんな狂気じみた幅広さなので，パソコンにまで興味が回らなくなってもしかたがない。

一般的にどんな人にでも共通していえることだが，アンテナを意図的に張っていないと，その分野に関する情報は入ってこない。いろいろな電波は飛んでいるが，チューナの周波数ダイヤルを合わせないとラジオが聞けないのと似ている。逆にアンテナさえ張れば，これだけ情報過多の時代であるから，あちこちから情報は飛び込んでくる。もちろん，そのまますぐ使える情報というのはほとんどない。だが探している情報のカケラが全然場違いの情報の中に隠れていたり，派生して別の形になって目の前に落ちていたりすることは多い。

たとえばプリンタの出物を目の色を変えて探しているときに，雑誌の広告の隅っこにポツンと載っている広告が目に飛び込んできたり，記事の比較表から突然インスピレーションを受けたりした経験が読者の皆さんにもあることと思う。

だから，情報収集においては，何よりもまず自分がどんな情報を求めているかを把握し，常に関連する情報に注意しようという意識を持つことが重要だ。漠然とそれを眺めていても，見逃がしてしまうものなのだが，アンテナを張っていると，あれこれとひっかかってくるから，いかにも人間の頭というか感覚は不思議にできている。

ところで，女の子を探す場合には，どうだろう。

「ガールフレンドなり婚約者を探している人は，自分の好みの女性のタイプをリストアップして，常にそのイメージに類似した女性を探せばよい」

という“恋愛必勝法”を何かで読んだ記憶がある。これはいままで書いてきた情報収集術とオーバーラップする。ということは，かなり的を射ているのだろうか？

確かに，論理的に考えれば，「女の子がいない」＝「次に好きになるべきターゲットを失ったまま」ということであるから，まずはターゲットを探す必要がある。よく少女マンガにあるような，突然のボーイ・ミーツ・ガールの美談なんて，まずないだろう。最近は恋愛映画でさえ，すでにつきあいだしている2人，という設定でドラマが始まってしまうことが多いのだから。

とはいえ，最初のとっかかりであるターゲットの発見が一番難しいのは，最近のご時勢ではいうまでもないこと。

X-OVER・NIGHT
（クロスオーバーナイト）

［第15話］
情報収集の妙味

TAKAHARA HIDEKI 高原 秀己

「某証券会社では手の届く範囲（つまり席が隣りや周り）にいる女性との社内結婚ばっかり」だの，

「某広告代理店では2回，チームとして仕事をしたら，必ず親しくなってしまう」

という神話に等しいような話をよく耳にする。ということは，それだけ今の世の中，出会いのチャンスが乏しくなっているといっていいのではないか？

ここで情報ピックアップの話に突然もどってしまうのだが，この初動段階での記事がほとんどないがゆえに，雑誌であきるほどやっている「真夏の恋愛大作戦100％必勝法」という記事は何の役にも立たないのである。あれほど無駄な記事というのもめずらしいのではなかろうか？

話題はクルクル変わる。エアコンがきかないほど暑くて，目が回っているかもしれない。最近，なぜか「東京スポーツ」（大阪では大阪スポーツ，九州では九州スポーツという名前で売られている）という夕刊新聞の話題がよく巷をにぎわせている。1面の奇抜な見出しを並べて作ってしまった単行本まで登場するありさまだ。

ぼくのようなプロレスフリークの人間にとってみれば，もう十数年，連日のように買い続けているから，不思議な存在の新聞ではないのだが，「新聞に書いてあることはだいたい正しい」と信じていた人たちに当たるも八卦，捏造もOK，というノリの1面編集方針がカルチャーショックを与え，それについて議論するのがブームになっているようなのだ。とくに，先日の「NHK島会長が女性と渡米」という記事が，結果的にスクープの一種になってしまったのが話題性に拍車をかけた。

とにかくウソ八百っぽい記事や見出しが圧倒的に多い新聞ではあるのだが，たまに当たるということは連日読んでいればわかる。見出しの右下に小さく「か？」がついていたり，2本の記事の見出しを組み合わせて，「まさか！」と思わせるような構成にしてあるのは，読んでみればわかる。このゲーム性に密かな楽しみを見い出しはじめたのが，ちょっとしたブームの正体といえる。こうしたゲームを楽しめるようになってきたということは，マスコミからの一方的な受け手である読者も相当グレードが上がってきたということだろう。

いかにも役に立ちそうなハウツーものの特集が何の役にも立たず，ゴミの山のような中にも情報がある。これを見抜くことこそが，情報収集の妙味といえようか？

ところで，今回の構成は，その昔，パソコンの使い方の本まで書いたぼくが，パソコンの使い方で頭を痛めている，という話に振って中盤から後半までを書く予定だったのだが（いちおう全体を3つくらいに分けて流していく基本テクは使って本稿を書いていますのでご安心を），残念ながら全然違う話へとスワップしてしまった。それだけ暑いということで，ご容赦されたし。この話は次回にしっかりと書くことにする。

愛読者プレゼント

1

システムソフト　☎092(752)5278

3名

キャンペーン版大戦略Ⅱ

X68000用　5"2HD版2枚組　9,800円(税別)

ウォーシミュレーションの大御所，大戦略シリーズの最新作。アクションはちょっと……，という人におススメです。

2

ファミリーソフト　☎03(3924)5727

装甲騎兵ボトムズ DEAD ASH

X68000用　5"2HD版2枚組

8,800円(税別)　3名

人気アニメのアクションゲーム化。ステージによって全方向スクロールになったり強制スクロールになったりするのがミソ。

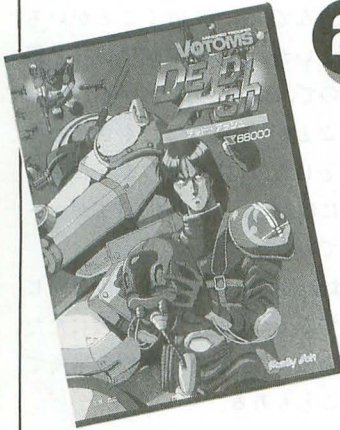
3

シャープ　☎03(3260)1161

ダッシュ野郎

X68000用　5"2HD版2枚組

8,800円(税別)　3名

手軽にできるバイクレースゲーム。とはいえ，整備されていない道もあったりしてなかなか難しい。サイバースティック対応。

4

アンス・コンサルタンツ　☎092(522)6347

サイクロンCGスペシャル

3,000円(税別)

3名

第2回サイクロンCG大会の入賞作品やCGテクニックを収めたビデオ。収録時間は20分と短いが，中身の濃いビデオだ。VHSのみ。

プレゼントの応募方法

とじ込みのアンケートはがきの該当項目をすべてご記入のうえ，希望するプレゼント番号をはがき右下のスペースにひとつ記入してお申し込みください。締め切りは1991年9月18日の到着分までとします。当選者の発表は1991年11月号で行います。

5

タヒボベビーダ

4名

ひさびさに登場の「清涼飲料水」。愛知県は千種駅で発見されたもので，提供者は山崎勘太郎さん。発売元はJR東海。味は……。

7月号プレゼント当選者

1ノスタルジア（北海道）佐藤博之（千葉県）秋葉貴男（愛知県）伊舎堂盛行（京都府）尾下克也（鹿児島県）木下良一　2シグナトリー（神奈川県）奥田健児　岡田啓介（大分県）石橋誠　3ボンバーマン（栃木県）戸辺靖（広島県）松本康裕（熊本県）本田幸裕　4イメージファイトテレカ（宮城県）郡山知行（栃木県）佐藤曜（埼玉県）渡辺一十六（東京都）栗原景　番場正和　5激変する半導体産業（東京都）池津英昭（神奈川県）吉田敏幸（大阪府）藍原和久（石川県）中村学（鳥取県）水船博一

以上の方々が当選されました。おめでとうございます。商品は順次発送いたしますが，入荷状況などにより遅れる場合もあります。また，雑誌公正競争規約の定めにより，このプレゼントに当選された方は，この号の他の懸賞には当選できない場合がありますのでご了承ください。

第62回

猫とコンピュータ

まだまだCOLUMNS

Takazawa Kyoko
高沢 恭子

単純明快なゲームほど，いつやっても気負わず楽しめるもの。キョウコさん一家は，いまあのブロックゲーム「Columns」にご執心のようです。しかし，のんびりとゲームをする間もなく，また新たな事件が……。

わずかな時間降った雨が，あたりを濡らし，空はまだひっそりと曇っている。アスファルトは黒く輝いて，向かいの団地を囲む木々の群れも，ひとときのうるおいにやすらいで見える。

フローリングの床をふき清めておくのはもとでのかからないわが家のゼイタク。盛夏はまだこれからだけれど，この上に戸外の揺れる光や，ホンニャアの動きまわる影を映すことで，涼感いっぱいの環境がたやすくできあがる。

その床をたどっていった先の開け放たれたドアの中から，「チャラン」「チロン」と，これも涼しそうな音が聞こえてくる。期末試験が終わって，夏の休暇に入る前になぜか試験休みのあるトオルが，マシンルームでひさしぶりに挑戦しているゲームの音だ。

うーん！ おもしろい

「Yet Another Column」とけしかけてくるブロックゲーム「Columns」は，'90年6月号の「Oh!X」創刊8周年記念の付録ディスクにおさめられていたものだ。

当時，「テトリスよりおもしろいや」と，ずいぶん楽しんでいたトオルだったが，このごろでは「Columns」ばかりでなく，ゲームをやることはあまりなくなっていた。それが先日，私がほんの気まぐれに手をつけてみたところ，これがまったくおもしろい。かつて編集部の人たちが，しごとを忘れて熱中したというのも無理はないと思った。

「トオル君，これおもしろいネー」

ルールはシンプル，目標はひとつ。だからたやすいかというと，そうはいかない。

初めて挑戦した私が大よろこびで低得点をあげているのを見て，トオルもあらためて新記録に挑みだしたというわけだ。きょう彼は50490点をあげた。

「Column」は柱。おもに柱頭，柱身，柱脚の3部から成る柱。あるいは縦隊，縦列，数字や項目が縦につらなったものをいうそうだ。新聞，雑誌の「コラム」も，このことだ。

たしかに3つの長方形（ブロック）でできた「柱」らしいものが，縦1列になって上から1本ずつ落ちてくる。3つのブロックは6色（赤，青，黄色，緑，紫，コバルトブルー）のどれかで組み合わされていて，落ちていく間に，縦の方向にだけ自由に組み変えられ，落下場所も9列の中から選ぶことができる。落ちたとき，縦，横，ナナメのいずれかの方向に，同色が3つ以上並んだら消去されて得点になる。消去できないものが積み上げられて，スペースが埋まってしまえばゲームオーバー。ただそれだけのことだ。

だがしかし，ルールがやさしいゲームの命題は，かならず速さだ。判断の早さ，指の動きの速さ，ミスのあとの処理の速さ。速さが得点を決める。

始めの5，6分はゆっくり，ゆっくり落ちてくる柱が，だんだん加速していって，終わりは息もつかせない。加速が始まるまでの間に柱のブロックをためこまないことに専念すれば，高得点の期待もある。でもそれも，1秒の慢心で夢は去ってしまう。

縦，横，ナナメに同色をそろえることは，回を重ねるごとにパターンを習熟していくから，指は機械的に働くようになるが，パターーンの選びかたのクセや，色に対する注目のしかたのちがいが，運命を無限に変えていくからおもしろい。

目玉と指の根気

下では，6つの色が，デコボコになった9つの列の中に散っている。はなやかでモザイクタイルのようだ。目はいつもこれらの色群の配置，配分を見ながら，上から来る3つの色を選別している。

どの色を選んで，どこに置くことがいちばん効率がよいか，この決定が早いほど得点があがるので，ほんとの血マナコになる。

しかもしごとが早いほど，たくさんの柱をさばくことになり，柱が矢のように降ってくるころには，指はともかく，目玉がヘトヘトになっている。

3つの色は全部同色だったり，2色が同じだったりもする。3つの色がどれも役にたたないこともあるが，どの色も複数の場所で有効ということもある。

そんなとき，慣れないうちは彩度の高い色のほうにひきずられがちになるが，トレーニング（？）されるうちに，均等な判断もじょうずになってくる。

色の並べかたのパターンは，じっさいにはたくさんあっても，一瞬の判断では基本的な部分を見るのがやっとなので，置いてみたら，予想した数よりたくさんのブロックが消去されておどろいたりする。

このうれしい誤算はわりあいひんぱんにやってくる。自分の判断こそ最善のつもりで，めざす場所に柱を置こうとするとき，落下点をあやまってしまうことがある。ところが，ミスしたつもりの場所で意外にもたくさんのブロックをつぎつぎ消去していくのだ。

自分が選択した色でない色が，偶然に同じ色と連結して，しかも連鎖的にひきおこされたナダレ現象から，一挙に空白のスペースをとりもどしてしまう。

この意図しない戦果が，「Columns」の魅力のひとつだ。小気味よい音を連続してあげながら，あれよあれよという間に，ブロックが崩れていく。じつは得点の何割かが，この偶然の力に助けられている。

私も，始めは2万点がやっとだったが，トオルに代わってチャレンジしたきょうの得

点は37030点，いままでのベストだ。

3万点の私と5万点のトオルのちがいはなんだろうと，彼のとなりにすわって観察してみたら，指の動きがくらべものにならないほど速い。1秒間に5回くらいキーを叩く。3万点を過ぎても残ったブロックは底のほうにわずかあるだけ。柱の落下がハイスピードになっても，落ちるまで距離があるからセレクトのゆとりがある。

でも，決定的なちがいは若さと根気。とちゅうで投げない，へばらない。やっぱりスポーツとゲームはこれで決まるらしい。

スコアのベストテンを登録する欄は，すべてトオルの名前で，10位が45350点，私の名前が入るのは不可能のようだ。

基地を新設

つゆ明けも間近の，ちょっとしのぎやすい日がつづいた日曜日の午後。

秋葉原の電気街を，小さなカートにシャープのFAXをのせた夫が，私の前になって歩いていた。ふと見ると，別のメーカーのFAXをかかえた男の人が，夫の荷物のパッケージをジーッとのぞきこみながら，同じ速度でついてくる。こちらの荷物には定価が貼られたままになっていたので，その人は自分が購入したものと，思わず比較をしていたのだろう。

夫のしごとの都合で，S市にある夫の家も，ビジネスの基地として使うことになった。水道，ガス，電気はすぐに使えるけれど，あとは家具が少し置いてあるだけなので，一応の生活用品をとりそろえなければならない。

そしてもちろんOA機器を完備しなくてはならない。これは少々手間も費用もかかるけれど，楽しみでもある。いままでの決して短くない経験から，もっとも使い勝手のよいシステムを，あたらしく設置するのだから。

そして，きょうは手始めに軽いものから，と，電話とFAXを仕入れたしだいだ。

FAXのシンプルな便利さには，1台めの「スピークス3」（NEC）ですっかり感服させられたのだが，少し残念なのは便利さを活用させる相手がまだまだ少ないということだ。

小さいころ，おとなのように電話をかけてみたくてたまらなかった。自分でダイヤルを回して，あの機械の向こうから誰かの声を聞いてみたい。両親のいないとき，わくわくして受話器をとりあげてみたが，おともだちの家にはまだ電話がひかれていないことに気がついた。

相手にそなえがなければ成り立ってくれないのが通信の世界だ。いくら便利な機械でも，ひとりで持っていることにはなんの意味もない。

パソコン通信の新しさにとりつかれて，FAXの明快なしごとぶりに気づくのがおそかったけれど，いざ活用しようとすると，こんどはなかなか相手のほうにそなえがない。ビジネスばかりでなく，友人や兄弟たちとも，明瞭で簡潔な連絡をしたいというときFAXは最適で，不必要な会話もしないですむから通信時間も節約になる。

「新宿も狛江もみんな入れたらいいのに」
「S市のおじいちゃんも，耳が少し遠くなったから，FAXがいいかもしれない」

パソコン通信とちがって，FAXなら誰でも使える。なんとかこの便利さを活用させたいと思っているとき，S市の家をOA装備することになった。

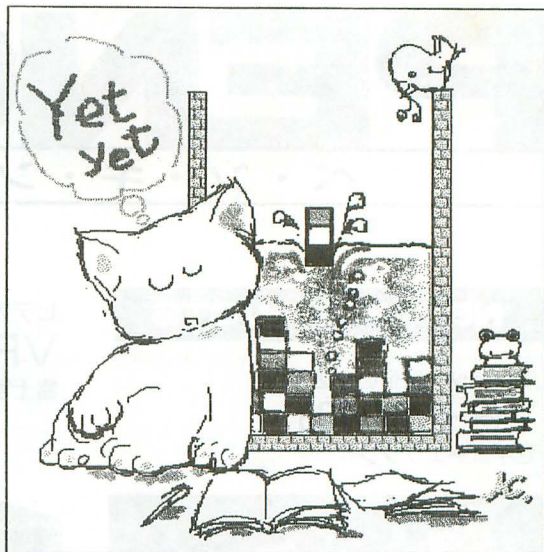
「FAXで連絡できるわね，パソコンでも通信できるし」と私がうれしそうに言うと，「あんまり遊んじゃダメだよ」と，夫が苦笑する。まわりにすすめる前に，自分たちでもう1台ふやしているのでは，やっぱり遊びに近いかもしれない。

新聞の投書や，テレビ番組のアンケートの回答などでも，このごろではFAX利用というのがふえてきた。それに，いま私たちの荷物を観察している人も，FAXを買ったところだ。きっと，これからおおいに活用しようというのだろう。

全体的な普及がすすまないと，通信は発展できないものだと，つくづく思う。

ディスクドライブの怪

「パソコンはやっぱり32ビットにしたいなぁ」と夫。

S市の家には最新のものを置こうということになり，つぎの日曜日も朝から計画をねり始めた。

「これも，もうフラフラだけど，当分は更新できないわね」

いま使っているPC-9801を，いたわるつもりでながめた。CRTはついこのあいだ，火を噴く様相を見せたために，あたらしくしたばかりだ。でもキーボードはすっかり貫禄がついた。

「とくに問題はないから，もう少し使ってやろうよ」

そんなやりとりをしながら，あれこれ購入の予定をたてていた。

「そういえば，またT店から，お中元用の通信カタログが届いてるのよ」

日本橋の老舗（しにせ）Tデパートから，パソコン通信で注文するためのディスクセットが，今回も送られてきていた。

「すこしは改良されたかな」

前回のトラブルの多さに，とてもこりてしまっていたが，半年間の進歩があるかもしれないと，夫はフロッピーディスクを差し込もうとした。

「あ？ あれ？」「どうしたの？」と，のぞきこむと，ディスクドライブの上段に入れようとしたフロッピーが，なぜか奥まで入らない。

何度やってもダメだ。何かがつかえている感じ，こんなことは初めてだ。水平に差し込んでいないのかもしれないと，ふたりで，10回以上もためしてみた。

ダメだとわかって，このまま修理に出すのは芸がない。何か，とても物理的な障害らしい。やっぱりあけてみようということになった。（以下次号へ）

PENGUIN INFORMATION CORNER

ペ・ン・ギ・ン・情・報・コ・ー・ナ・ー

NEW PRODUCTS

ワンタッチ用紙切り替え
VP-870/1700
セイコーエプソン

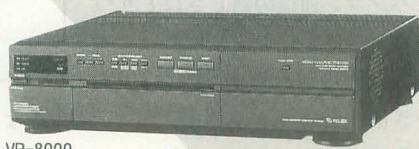
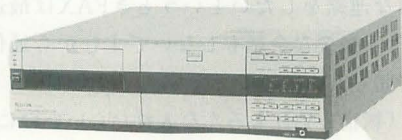
セイコーエプソンは，スーパーマルチウェイローディング機能搭載の24ピンインパクトドットマトリクス漢字プリンタ「VP-870/1700」を発売した。

スーパーマルチウェイローディング機能とは，従来の単票用紙と連続用紙の切り替えがワンタッチで行えるマルチウェイローディング機能をグレードアップしたものである。

本体のフロント・リア・トップ・ボトムの4方向から給紙するマルチパスシステム。トラクタを2基標準装備し，2種類の連続用紙をフロントとリアにセットし，レバー操作で切り替えを可能にしたデュアルペーパーパーク。トラクタのセット位置をフロント・リア・トップ（プル給紙用）に選択できるマルチトラクタシステム。以上で構成されているこの機能により，使用環境に応じて給紙の方向を自由に変更，選択ができ，また，オプションのカットシートフィーダを装着すると，最大4種類の用紙（2種類の連続用紙と2種類の単票用紙）を常時セットすることができる。コントロールコードはESC/P24-J84を装備。

価格は136桁の「VP-1700」が166,000円，80桁の「VP-870」が136,000円（いずれも税別）。

〈問い合わせ先〉
エプソンインフォメーションセンター
☎03(3377)3500, 06(212)8712, 052(953)9239

ビデオプリンタ
VP-8000/8100
富士写真フイルム

VP-8000

VP-8100

富士写真フイルムは，昇華型熱転写方式の高画質フルカラービデオプリンタ「フジックス ビデオグラフィックプリンタ VP-8000」，およびその上位機種「VP-8100」を発売した。

この「VP-8000/8100」は，色ノリがよく原色に忠実な色再現性を実現するために，新開発の高画質ペーパーインクセットに最適な画像処理回路を搭載。シアン，マゼンタ，イエローの発色波長バランスを最適にすることにより，従来ビデオプリンタでは難しいとされていた肌色の再現性も向上させている。

静止画像だけでなく，ビデオカメラ，VTR,テレビ放送などの動画からも任意に選んだ1シーンを内蔵メモリが記憶し，プリントできる。さらに，内蔵メモリに記憶した2つの画像を1枚の受像紙にプリントすることもできるので，経時変化などの比較にも便利。

これに加えて，上位機種の「VP-8100」では以下のような機能を備えている。水平周波数が15～32kHz（垂直周波数45～70kHz）までのRGB信号をメモリできる，オートスキャン機能。異なる画像を4，16，25，64に分割してメモリし，1枚の受像紙にプリントできる，マルチメモリ機能。プリントエリアの設定。

入出力端子はビデオ端子，S映像端子，BNCジャック，アナログ/デジタルRGB信号（「VP-8100」のみ）といった豊富な入出力端子を備えている。

価格は「VP-8000」が198,000円，「VP-8100」が450,000円で，専用のペーパーインクセットが3,500円（「VP-8000」用，50枚），9,000円（「VP-8100」用,100枚）となっている（すべて税別）。

〈問い合わせ先〉
富士写真フィルム(株)　☎03(3406)2111

高精細RGB液晶プロジェクタ
CU-SX1
シャープ

CU-SX1

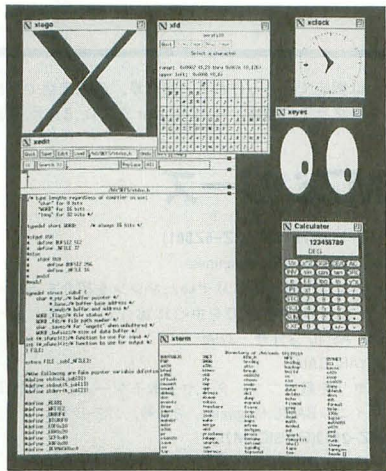
シャープは，パソコン画面を最大150型の画面サイズで表示できる液晶プロジェクタ「CU-SX1」を発売した。

この「CU-SX1」では新開発の高精細液晶パネル(21万8千画素)を3枚使用，合計65万画素という高画質化を実現している。こまやかな文字や図形などの表現能力を高めているので，パソコンを使っての大画面プレゼンテーションなどに向いている。ビデオ端子やS端子に入力した映像に，RGBの画像をスーパーインポーズすることもできる。

入力は水平周波数15kHzのアナログ信号に対応。また，水平周波数24kHz映像出力を水平周波数15kHzに変換するスキャンコンバータ「XC-10SC1」も同時発売される。

価格は980,000円。スキャンコンバータ「XC-10SC1」は300,000円(ともに税別)。

〈問い合わせ先〉
シャープ(株)☎03(3260)1161, 06(621)1221

OS-9用X Window
Microware/X Windows
マイクロウェア・システムズ

マイクロウェア・システムズは，MITのX Window(Version11 Release4)をOS-9に移植，「OS-9/X Windows」を発売した。

X Window Systemはソフトウェアやハードウェアに依存せず，多様なパーソナルコンピュータ，ワークステーション，およびメインフレームコンピュータで使用できるグラフィックウィンドウシステムである。そして，この「OS-9/X Windows」ではこれまでのX Window Systemでは不可能だったROM化も可能になっている。

製品は以下の4つのパッケージに分かれていて，その用途に応じた購入ができる。

OS-9/X Windows Client Development Pak

OS-9システム用のX Windowアプリケーション開発用ツールセット。400,000円。

OS-9/X Windows Client Run Time Pak

手持ちのOS-9/X Clientアプリケーションを OS-9ターゲットシステム上で展開するために必要なすべてのツールを含んでいる。60,000円。

OS-9/X Windows Full Source Pak

マイクロウェアがOS-9上で動作するように変更を加えたMIT X Window Systemのすべてのソースコードを含んでいる。このパッケージはマイクロウェアの移植したX Windowsを変更したり拡張したりするときに使用する開発者向けのパッケージである。開発中。

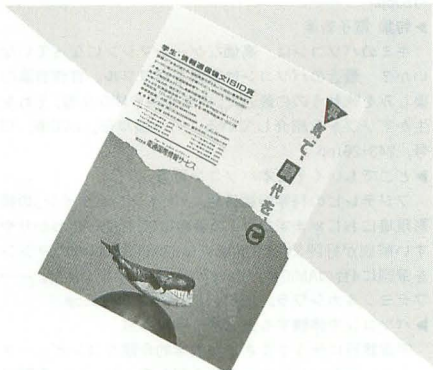
OS-9/X Windows Server Sample Source Code Pak

MIT X Windowサーバ・ソース・コードすべてと，OS-9/Xサーバをサポートするために必要な情報が含まれている。このパッケージは既存のOS-9/レジデントまたはクロス開発システムにインストールするためのもの。開発中。

供給メディアは1/4inchカートリッジMTで，マイクロウェアからの直販のみ。

〈問い合わせ先〉
マイクロウェア・システムズ㈱
☎03(3257)9000

INFORMATION

大学生を対象とした
学生・情報通信論文ISID賞設立
電通国際情報サービス

電通国際情報サービスは大学生を対象とした「学生・情報通信論文ISID賞」を新たに設立，募集を開始している。

この「学生・情報通信論文ISID賞」はISIDの創立15周年記念事業の一環として設立されるもので，次代を担う大学生に情報通信への期待，要望，夢を語る機会を提供することによって，情報通信のあり方を探ることを目的としている。1991年を第1回として今後毎年1回募集していく予定になっている。

応募課題は「未来が変わる。情報通信が変える」と，「自由課題」(ただし，情報通信に関連したもの)の2テーマとしており，生活レベルの発想から専門的なものまで幅広く応募を求めている。

審査委員は，委員長の電気通信大学の合田周平氏をはじめ，国立民族学博物館助教授の石森秀三氏，経済評論家の日下公人氏，漫画家の杉浦日向子氏，そして，後援団体である財団法人テレコム高度利用推進センター理事長の成川富彦氏が決定している。

●応募要領
対象　日本国内の大学に在籍する学生個人，および，グループ
規定　A4判，400字詰め原稿用紙10～20枚程度。ワープロ使用の場合はA4判1行40字×20行で，5～10枚程度。いずれも横書き。ただし，表紙，目次，付表，付図，参考文献，注釈などは含まず。なお，これらもA4判にまとめること。
締め切り　1991年9月30日(月)必着
応募先　〒164 東京都中野区中野4-11-10
　株式会社電通国際情報サービス 学生・情報通信論文ISID賞事務局係

〈問い合わせ先〉
㈱電通国際情報サービス　☎03(3228)6100

コンピュータ技術者養成
ヒューマン・クリエイティブ・スクール

ヒューマン・クリエイティブ・スクールでは，コンピュータ産業が必要とする優れた人材を育成するための実務教育と先端技術を探求し，特に専門的にコンピュータゲーム制作技術者を養成している。

同校ではソフト開発にかかわる人材不足の解消などを理由に，現在の1年制から2年制への変更を行う。

さらに，ゲーム産業とAV産業が同化し発達していくなかで，通常のゲームグラフィックとは異なる企画，脚本，キャラクタ，美術設定，プロデュースなど，主にストーリーゲームを視野に入れたコンピュータグラフィックを専門に教育するコースも新設した。

〈問い合わせ先〉
ヒューマン・クリエイティブ・スクール
☎0422(22)1171

SAPPORO CG'91
北海道コンピュータグラフィックス協会

北海道コンピュータグラフィックス協会は1987年2月に北海道におけるCGの研究促進，技術向上，ビジネスや産業への利用発展などを目的として設立された。この協会のメインの活動のひとつである「SAPPORO CG'91」が今年は11月26日～11月28日までの期間，札幌パークホテルにて開催される。このイベントは道内で唯一のCGに関する大規模なイベントであり，CG技術に関連する企業や研究機関が展示，発表を行い，また，内外の著名人による研究発表会，トークショウなどが行われる予定。

また，「SAPPORO CG'91」のもうひとつのイベントとして，「第4回CG作品コンクール」も開催され，プロから一般マニアまで幅広く作品募集を行っている。締め切りは11月19日(火)まで。詳しい応募要領の問い合わせや，応募用紙の請求は事務局まで。

〈問い合わせ先〉
北海道コンピュータグラフィックス協会
☎011(232)3330

FILES Oh!X

このインデックスは，タイトル，注記——筆者名，誌名，月号，ページで構成されています。お盆も終わって，長かった休みもそろそろ終盤。思い残すことがないよう，有意義に過ごしましょうね。

参考文献
I/O 工学社
ASCII アスキー
コンプティーク 角川書店
テクノポリス 徳間書店
ポケコンジャーナル 工学社
POPCOM 小学館
マイコン 電波新聞社
マイコンBASIC Magazine 電波新聞社
LOGIN アスキー

一般

▶特集!! 同人ソフトサークルは，なぜ同人ソフトを作り続けるのか!?
毎週どこかで行われているといわれるパソケット（パソコン用同人ソフト即売会）。参加する若者たちは何を目的に集うのか？　また同人サークルとはどんなものなのか？　をさぐる。——編集部，テクノポリス，8月号，147-150pp.

▶特集 電子音楽
キミのパソコンは，高価なゲームマシンになっていないか？　最近のパソコンは音源もパワフル。自作自演の楽しみを味わうのに最適だ。内蔵音源やMIDI楽器，それを生かすソフトを紹介している。——編集部，LOGIN，13号，243-261pp.

▶どこでもいくぞ日本パソコン百景
フジテレビの科学解説番組，「アインシュタイン」の撮影現場におじゃまする。この番組はCGを使ったわかりやすい解説が好評で，CG作成には68030の50MHzのマシンを筆頭に4台のAMIGAが使われているということだ。——フデヨシ＆カシワラ，ASCII，8月号，234-235pp.

▶パソコンで体験する天文学・宇宙の旅
宇宙旅行に伴うさまざまな数学的命題をコンピュータシミュレーションによって解き明かすシリーズ。今回は宇宙旅行の全行程の距離・所要時間を算出してくれるプログラムと，HR図で恒星の進化経路を見せてくれるプログラムの2つ。——福江純・藤原隆男，ASCII，8月号，302-307pp.

▶対決！　シリーズ第3弾 MicrosoftExcel vs. Lotus 1-2-3
新人ビジネスマンと先輩OLの会話形式で，代表的な表計算ソフト2つの特徴を調べ，比較するコーナー。——根岸邦彦，ASCII，8月号，325-332pp.

▶欧州ハイテク事情
イタリアに滞在している筆者が贈るイタリアコンピュータ事情。秋葉原のような電気街はないが，カタログ販売サービスが発達しているのが救いだとか。——菊地薫，ASCII，8月号，396-397pp.

▶MYCOM WATCHING
大東文化大学就職部に採用されているコンピュータデータベースを取材する。企業情報やOBの情報を簡単に引き出せ，地域別のリストアップなども行える。Uターン組には特に好評とか。——菊地秀一，マイコン，8月号，230-233pp.

▶'91東京おもちゃショー
幕張メッセで開かれたおもちゃショーの模様を豊富な写真入りで紹介する。——編集部，マイコン，8月号，336-337pp.

▶超簡単オシロの製作
低機能ではあるけれど，超簡単，超低価格で作れるオシログラフの製作。——Dreamer.Y，I/O，8月号，194-195pp.

MZシリーズ

MZ-1500(BASIC MZ-5Z001)

▶Saving All Our Machines
今では一線からしりぞいたマシンを紹介するコーナー。MZ-1500の特徴をQDを中心に紹介している。——編集部，ASCII，8月号，375p.

▶YARI GAME
ヤリを避けつつ，ダイヤを取るゲーム。——竹田雄介，マイコンBASIC Magazine，8月号，122-123pp.

MZ-2500(BASIC-M25)

▶Illegal Mission
ジョイスティック専用のシューティングゲーム。——アダモ，マイコンBASIC Magazine，8月号，124-126pp.

X1/turbo/Z

X1シリーズ

▶JEWELRY
宝石を積んでいくアクションパズルゲーム。——岡本公功，マイコンBASIC Magazine，8月号，148-149pp.

▶HEARTEN
吐き出された心を集めるのが仕事。固定画面の弾よけアクションゲーム。——渡辺剛，マイコンBASIC Magazine，8月号，150-151pp.

▶うちのタマしにませんか
障害物をよけ，壁に反射させながらゴールへ導くアクション迷路ゲーム。——堀田英克，マイコンBASIC Magazine，8月号，152-153pp.

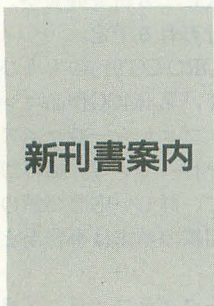
▶グラフィックSAVE&LOADユーティリティ
その名のとおり，X1とturboのグラフィックをセーブ，ロードできるユーティリティ。圧縮も白黒変換もしてくれるぞ。——坊農誠，マイコン，8月号，365p.

X1＋FM音源ボード（要NEW FM音源ドライバ）

▶エメラルドドラゴン
ゲームミュージックプログラム。——KENJI，マイコンBASIC Magazine，8月号，177-178pp.

X68000

▶GAMING WORLD
3Dシューティング「装甲騎兵ボトムズDEAD ASH」や，パズル「スターモビール」などを紹介。——編集部，テクノポリス，8月号，24-28pp.

▶SOFT EXPRESS

かなりややこしいタイトルだが，内容は「都市論」だ。しかも，東京都心部に見られる「その日暮らし」型生活という，従来支配的であった“我が家は安住の地”ともいうべき持ち家論とは正反対の生活を見ながら，「過程（プロセス）に生きる」と大胆な結論を導きだす。都市に住むということを前提にすると，住居・職場・ショッピングが渾然となった都市での生活は，常に「完成」されることはなく「過程」の中にあることを悟ろうというわけだ。
著者は近未来の都市をそう描く。人工的で，コンパクトで，立体的な近未来の都市。著者はそういった都市での生活を提示しながら，ときには歴史的な建築物や都市を例に挙げて専門的に解説しながら，都市の作りそのものを見せる。それはローマの水道橋と東京の首都高の対比であったり，川を中心としたアン・アントニオの見事な都市作りであったり，著者が参加した「つくば博」が暗示する「消滅を前提とした都市」だったりする。
つまりは，人工的でブレードランナーな未来に向かって，前向きに係わっていこう，ということだ。同時に，都市設計家の著者が今の東京をどうしたいのかが見てとれて面白い。（K）

未来史の脳人都市　山田雅夫著　河出書房新社刊
☎03(3404)1201　四六判　245ページ　2,200円

書籍の価格は消費税込みです

生中継68，装甲騎兵ボトムズ DEAD ASH，アークス・オデッセイ，ゼノン2，オルテウス2などを紹介。――編集部，コンプティーク，8月号，66-75pp.

▶Software Hot Press

8月発売予定のアクアレスや，グループ・エックス，生中継68，イースなどの新作ゲームを紹介。――編集部，POPCOM，8月号，12-23pp.

▶ゲームの達人

話題のゲーム，ファランクスを攻略。――編集部・ボンボコリン後藤，POPCOM，8月号，92-93pp.

▶NEW SOFTWARE

SX-WINDOW初のアプリケーションソフトである，16色グラフィックツール「Easypaint SX-68K」と，ウィンドウ環境を取り入れた高機能ワープロ「Multiword PRO-68K」を紹介している。――6st.Inv.，マイコンBASIC Magazine，8月号，76-78pp.

▶誌上公開質問状コーナー

スプライトエディタやグラフィックツールで作ったデータを，X-BASICで利用するにはどうすればよいか？などの質問に答える。――多田太郎，マイコンBASIC Magazine，8月号，99-100pp.

▶夕暮れの冒険

障害物をよけまくるアクションゲーム。――沢田広正，マイコンBASIC Magazine，8月号，154-155pp.

▶マリ夫の冒険

雲の上を自由に飛び回るマリ夫。固定画面のジャンプアクションゲーム。――高橋秀之，マイコンBASIC Magazine，8月号，156-158pp.

▶PRACTICE

コースが動く3Dタイプのカーレースゲーム。――長谷川光助，マイコンBASIC Magazine，8月号，159-161pp.

▶驚異の頭脳開発テストPRO68K

クイズソフトのメインルーチンに使える。対戦クイズもできる。PC-8801からの移植。――宮城良行，マイコンBASIC Magazine，8月号，167-168pp.

▶ファミコン版悪魔城ドラキュラ 〜バンパイアキラー〜

コナミのゲームミュージックプログラム。要NAGDRV＋CM-32L系MIDI楽器。――貞満政紀，マイコンBASIC Magazine，8月号，179-181pp.

▶今月の注目ソフト

X68000ユーザー待望のイースがついに移植される。もちろん，X68000の機能を十分生かした作品だ。ほかに生中継68の紹介。――佐久間亮介，マイコンBASIC Magazine，8月号，239-244pp.

▶NEW SOFT

NEW 3D GOLF SIMULATION用オリジナルコースデータ「エイトレイクス・ゴルフクラブ」を紹介。――編集部，LOGIN，13号，23p.

▶最新ゲーム徹底解剖!!

ファランクス徹底解剖の最終回。6面，7面を攻略。――編集部，LOGIN，13号，180-181pp.

▶Software Review オリジナルゲームの重要性はいかに？

ファランクスをネタに，オリジナルゲームの楽しさを探る?!――X68000新聞社，LOGIN，13号，208-209pp.

▶X68000新聞

パソコンフォーラム'91の模様。SX-WINDOW用グラフィックツール「Easypaint SX-68K」やゲーム「アークスオデッセイ」「生中継68」「オルテウス2」など新作情報。――編集部，LOGIN，13号，276-279pp.

▶NEW SOFT

リアルなグラフィックの野球ゲーム「生中継68」を紹介。――編集部，LOGIN，14号，16p.

▶X68000新聞

「インペリアル・フォース」「3D2（仮題）」「イース」といった新作情報と，第1回全日本X68000芸術祭のお知らせ。――編集部，LOGIN，14号，250-251pp.

▶AVプログラミング講座

ポリゴングラフィックス・プログラミングも今回が最終回。まとめとして，Zバッファ法などさまざまな種類のアルゴリズムや実際にプログラミングするうえでの処理の高速化について解説する。――宮本親一郎，ASCII，8月号，341-348pp.

▶AV STRASSE

X68000のSX-WINDOWに対応したペイント系グラフィックエディタ「Easypaint SX-68K」が紹介されている。イメージスキャナ/プリンタ用のアプリケーションがそれぞれ付属。――編集部，ASCII，8月号，357-360pp.

▶長期ロードテスト

FIX68000EXPERT試用記第7回。ゲームを試しての感想と，FIXERへの登録単語の移し替え，SX-WINDOW Ver.1.1の環境についてなどで，評価はいずれも高かった。――編集部，ASCII，8月号，416-418pp.

▶Mycom Soft Review

シャープから発売されていたCARD PRO-68KがVer.2.0にバージョンアップされた。このページではデータベースとは何かから始まって，一連の作業の模様とその使い勝手をレポートしている。――都築敏也，マイコン，8月号，134-139pp.

▶LET'S PROGRAM

今月の宿題は「途中に小数点が入る最大20桁×20桁の掛け算」。読者からの解答ではX68000上のGNU Cで書かれたプログラムが載っている。――藤本健，マイコン，8月号，173-181pp.

▶HOBBY EXPRESS

X68000シューティングゲーム「スコルピウス」と超大作アドベンチャーゲーム「シグナトリー」を紹介している。そのほか，A列車で行こうIIIの徹底研究など。――あゆさわかつみ・成島月実ほか，マイコン，8月号，339-363pp.

▶NED

多機種対応のフルスクリーンエディタ。コンパクトで移植性に優れているのが特徴で，コンソール制御は原則としてエスケープシーケンスで行う。――ネコ老師，I/O，8月号，80-95pp.

▶GAME BOX

コムパックの「グループ・エックス」，シャープの「ダッシュ野郎」，そして工画堂スタジオの「サブナック」の３本が紹介されている。――YRK'sほか，I/O，8月号，106-110pp.

▶DB-Z プログラムBASICカード

シャープから8月に発売されるDB-Z専用のプログラムBASICカード，PA-9C95/9C96の概要をレポートする。――塚田洋一，ポケコンジャーナル，8月号，4-10pp.

▶電子手帳とパソコンをリンク

電子手帳の欠点である日本語入力の弱さをカバーするために，HAL研究所の「Hyper HAL-CATCH」を使ってDB-Zとパソコンとをつないで活用しようという記事。――松田ぱこん，ポケコンジャーナル，8月号，11-16pp.

ポケコン

PC-G801/802/803/811/813/PC-E200

▶ポケコンで制御の実習をしよう

「Z80システムバス」にインタフェイスボード「8255ボード」をつなぎ，Z80CPUのマシン語を利用して機械などの制御方法を学習する。――太平洋工業・ポケコン研究会，ポケコンジャーナル，8月号，43-47pp.

PC-E500/E550/1480U/1490U

▶パズルゲームなんてクソくらえだ！

薬品同士を混ぜ合わせて新しい薬品を合成するパズルゲーム。――Abu Soft.，ポケコンジャーナル，8月号，72-73pp.

▶2次方程式の求根

整数係数の2次方程式の解を求めるプログラム。解の公式ではなく，人間的なアルゴリズムで求めてくれる。――Min，ポケコンジャーナル，8月号，86-87pp.

▶RETURN

パズルを組み替えてボールを導く。チクタクバンバンみたいなゲーム。――Mr.Sheep，マイコンBASIC Magazine，8月号，163-164pp.

PC-1360K

▶PC-Editor

プログラム編集用のフルスクリーンエディタ。リロケータブルで，自分で欲しい機能を付加できるのがウリ。――K.M，ポケコンジャーナル，8月号，88-93pp.

デジタル・ナルシス

コンピュータ文化論である。各章の副題が面白いので，すべて書いてしまおう。ジョン・フォン・ノイマンのユートピア，アラン・チューリングのエロス，チャールズ・バベッジのロマン，クロード・シャノンのダンディズム，グレゴリー・ベイトソンのアクロバット，ノーバート・ウィナーのクルーセイド，機械は第三の性。だいたいどんな内容かわかってくると思う。興味を持った人は一読を。（K）

西垣通著　岩波書店刊　☎03(3265)4111
四六判　235ページ　2,100円

ザ・ベストゲーム

いままでに登場したすべてのビデオゲームから読者投票でベストゲームを決めた。とはいえ，月刊ゲーメストの読者層を考えればわかるとおり，かなりマニアックなゲームや新しいゲームが上位に入っている。なんと１位が「ワルキューレの伝説」。そのあたりがやや残念だが，ビデオゲームフルリストは必見。特に，懐かしいゲームに郷愁を覚える。ちなみに，私のベストゲームは「クレイジー・クライマー」。（K）

月刊ゲーメスト増刊　新声社　☎03(3293)9321
AB判　240ページ　1,980円

書籍の価格は消費税込みです

QUESTION and ANSWER

Oh!Xに掲載されている音楽プログラムはOPMDやMUSICDRV用のプログラムが多くの曲を占めていて，音楽プログラムだけを打っても聞くことができません。音楽プログラムを掲載するときにOPMDやMUSICDRV，ZMUSIC.FNCのプログラムリストを一緒に掲載してください。友人もまったく同じような状態でたいへん困っています。お願いします。　広島県 長尾圭悟

6月号に掲載されていたミュージックプログラムについてですけど，リスト2のブルーウォーターのプログラムを入力して実行したんですが，エラー(外部関数エラーです。Yの指定に誤りがあります……3700行）が出てきました。どこか間違えているのではないですか。それともうひとつコンフィグファイルの使い方を教えてください。

宮崎県 秋田直人

音楽関係の質問を集めてみました。まずは長尾さんのほうからいきましょう。MUSICDRVの掲載された1991年5月号はまだバックナンバーで入手可能です。MUSICDRVのデータはほとんどがMIDI対応ですから，Musicstudio PRO-68K Ver.2.0やMu-1 Superを購入されるのもよいでしょう。

ZMUSIC.FNC用のデータはZMUSIC.FNC発表時のサンプルを入れても2つしか発表されていません。素直にあきらめてください。ZMUSIC.FNC用のデータはZMUSIC.FNCがなければ演奏できませんので，今後ZMUSIC.FNC対応のデータが掲載されることはないと思われます。

残るはOPMDですがOPMDやOPMA対応のデータでは長尾さんのおっしゃるような「プログラムを打ち込んでも聞くことができない」という事態はほとんど起こりえません。

リスト1

```
          ⋮
3680 func m_trns(x)
3690   for i=0 to 34
3700   m_trk(x,a(i))
3710   next
3720 endfunc
          ⋮
```

OPMDが発表されてすでに1年半，その前身のOPMAからは2年近くたちました。初心者の方には単に「OPMD対応」といってもなんのことだか理解できないはずなので解説しておきます。

OPMDとは基本的にFM音源とAD PCMを同期して扱えるようにするためのプログラムです。音楽の途中でドラムを鳴らしたりするために使われます。OPMD用のデータはOPMDのないシステムでも問題なく使用できるように，巧妙な方法でAD PCMを制御しています。それはYコマンドというFM音源制御命令のうち，実際には使用されていない部分に機能を割り当てるというかたちで実現されています。そしてメモリ上に読み込んだAD PCMデータをMMLで指定したときに鳴らすことができるように，FM音源へ送られるデータを見張っているのです。

よってOPMD用のデータをOPMDなしで使用してもドラムが鳴らないこと以外，なにも問題はありません。OPMDやMUSICDRVまたは同等品を入手すればドラムは鳴るようになりますから，いまのうちからデータを蓄積しておくことは無駄ではないでしょう。

また，毎月OPMD本体をダンプリストで掲載することは可能ですが，これだけではまったく無意味です。問題なのはサンプリング音のほうなのですから。もちろん，毎月数10Kバイトのダンプリストを打ち込みたいというのなら話は別ですが……。

サンプリングデータを収録した1990年6月号のバックナンバーが切れて久しく，Oh!X LIVE in '91で使用されているサンプリングデータを入手できないという人も増えています。ほとんど圧縮のきかない400Kバイトのデータですから付録ディスクのたびごとに収録するのも大変です。すでに対策のプロジェクトは進行中ですので，もう1，2カ月のあいだお待ちください。

続いて秋田さんのほうにいきます。念のため掲載したデータをもう一度確認してみましたが，すべて間違いなく入力されているのならちゃんと動くはずです。

ミュージックデータは入力ミスが発見しにくいことで有名なのですが，そんなときこそBASICの対話的環境が生かされます。このプログラムは3700行でm_trk( )をまとめて行っているため，どこで間違っているのかわかりにくくなっています(リスト1)。

しかし，この例ではxとiの値がわかれば，どのデータ列でエラーが起こったかが特定できます。エラー発生直後にxの値を表示させてみればどのトラックでエラーが起きたかがわかります。ここでは変数iがローカル変数となっているので，そのままではエラー発生時の値がわかりません（ローカル変数は保護されない)。しかたないのでエラーが出なくなるまで，

3695 print i

のようにするしかないでしょう。あるいはもっと安直に，

3695 print x, i, a(i)

とすれば嫌でも間違えた場所がわかるはずです（画面がうっとうしくなりますが)。

また，OPMDのコンフィグファイルですが，主にサンプリング音を登録するために使用されます。ですからOPMD単体ではなく，1990年6月号のディスクで収録したようなサンプリングファイルが必要な場合がほとんどですので注意してください。

OPMD用のコンフィグファイルはエディタ上で作成します。適当な名前でセーブしたなら，OPMD起動時に，

OPMD ADPCM.CNF

のように指定します。

このファイルの内容は，ドラムの何番にどの音を割り当てるか，が基本になっており，たとえば，

10=¥SAMPLING¥KICK.PCM

という指定は「10番にSAMPLINGディレクトリにあるKICK.PCMというファイルを割り当てる」という意味になります。サンプリング音がどこにあるかは人それぞれで違うので，「=」以下のパス名の部分は各自のシステム構成にあわせて変更するようにしてください。

なお，そのほかにもOPMDにはデータをまったく変えずにMIDIへ出力ができるという利点もあります。その際の音色変換でもコンフィグファイルは使用されます。内蔵FM音源の音色配列とOPMD標準のボスコニアン準拠のドラム配列に各MIDI楽器の音色配列を対応させることで，手軽にMIDIを扱えるようにしたものです。OPMDをOh!X LIVEのみで使用するなら，こちらはあまり気にしなくてもいいでしょう。1990年6月号で同時に収録していたコンフィグファイルを参考にしてください。

CPUからVRAMをアクセスする際に表示側の回路とアクセスがぶつからないようにしなければいけませんが，具体的にはどのようにやっているのでしょうか。CPUがアクセスしてもいい期間とそうでないときのCPUの待たせ方などについて教えてください。機種はMZ-700です。　神奈川県 石田 伯仁

VRAMというのはCPUだけでなくCRTCなどの表示コントローラからもアクセスされる特殊なメモリです。もし，CPUとCRTCが同時に同じところをアクセスすると（バスがぶつかると）非常にまずいことになります。

それを回避するために一般的に行われていた方法は，CPUアクセスのタイミングと表示側のアクセスするタイミングを完全に分離することです。CPUがごねてはいけません。どちらが優先されるべきかというと表示側です（表示がおかしくなりますからね）。そのため，CPUはVRAMアクセスの際は必ず1ウェイトを入れてCRTCのアクセスしていないタイミングでVRAMを操作していました。

これでは効率が悪いので，VRAMアクセスのタイミングを2倍に分割し，見掛け上，同じクロック時間で両者がアクセスできるようにした方式がサイクルスチールと呼ばれるものです。昔，FM77で採用されて話題になった方式ですね。X1シリーズでもこれと同様のアクセス方式をとっています。

しかし，この方法はクロックが速くなると通用しなくなります。そのためX68000などでは同時に2カ所からアクセスされてもRAM側で対処できるデュアルポートDRAMが使用されています。

MZ-700ではあまり縁のない話かもしれませんが，元がシンプルでハードウェア派も多いMZ-700です。2HDドライブやハードディスクを装備……という話も耳にしています。ROMをノーウェイトの高速版にして，VRAMをデュアルポートにして，CPUも64180Zかなにかにして……というマシンもどこかにあるかもしれません。

（中野　修一）

いまX-BASICの外部関数を作成しているのですが，デバッグ作業が思いどおりにいかないので困っています。なにかよい知恵はないでしょうか。　和歌山県 大西 則幸

プログラムの開発はプログラミング/デバッグの繰り返しです。ソースリストの字面をいくら眺めてみても原因のわからないバグにぶつかることもよくあることです。たいていはプログラマの思い込みが原因です。第3者が見ればすぐにでもわかるような間抜けな間違いほど，本人にはわかりづらいものなのです。そんなときデバッガはプログラマの強い味方となってくれます。いいプログラムを効率よく開発するには，いいデバッガが必要です。私もずいぶんとデバッガにはお世話になっています。

さて，通常の＊.Xファイルはデバッガにファイルを読み込んで，起動前にブレイクポイントを設定することができるのですが，X-BASICの外部関数だと話がややこしくなってきます。

　A>db a:¥basic2¥basic.x

として，デバッガにbasic.xを読み込んでも，このとき外部関数はメモリに読み込まれていません。ここでGコマンドを使えばX-BASICは外部関数を組み込んで起動します。しかし，目的の外部関数がどこにロードされているかはなかなかわかりません。さて，ではどうすればいいでしょうか。

私も外部関数のデバッグにはずいぶん悩まされました。悩んだ末に思いついた方法が，トレースしたいプログラムの直前にわざとアドレスエラーを発生させる命令コードを置く方法です。たとえば奇数番地をワードでアクセスするようにするとかですね。このような仕込みを加えた外部関数をX-BASIC（BASIC.CNF）に読み込んで，デバッガを起動します。仕掛けたコードを実行しようとすると例のエラーウィンドウが現れるかわりに，デバッガに制御が移りますから，ここから1命令ずつトレースしていくようにすればいいでしょう。この方法は結構使えますが，アドレスエラーを発生させるというやり方が気にいりません。

たとえばZ80ならブレイクポインタを指定すると，指定アドレスにRST 38Hが書き込まれます。これはCALL 38Hと同じことで，この番地を呼ぶとモニタに制御を移すようになっているのです。これと同じような命令が68000にあればいいのにと思っていました。

そんな，ある日プログラマーズマニュアルをなにげなく眺めていたら，なんとTRAP #9でZ80のRST38H相当の動作をすることが書いてあるではないですか。TRAP命令はIOCSコールでも使っているので皆さんもお馴染みのものですよね。この命令は68000の例外処理で，基本的なサブルーチンの呼び出しに使われます。市販のゲームソフトにも，TRAP命令を使って音楽演奏をするものがありますね。

これは便利，と思い，さっそくソースリストに挿入してX-BASICを実行すると，

　エラー($0029)が発生しました

とメッセージ。それもそのはず，デバッガからX-BASICを起動しないとこうなってしまうのです。デバッグ時のみ使用してください。

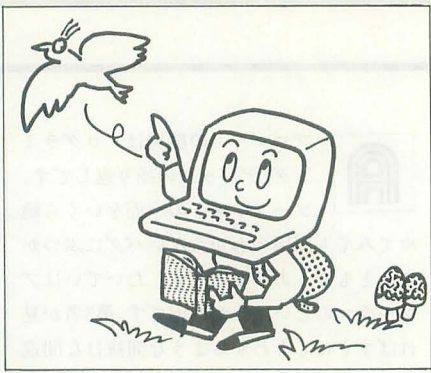
そのほかにも，外部関数の先頭に文字列を埋め込み，デバッガでサーチするという手もあります。もちろん，文字列部分は非実行部分にしておきます。これならデバッガを使わずBASICだけで起動しても大丈夫ですが，デバッグのたびにサーチの範囲指定とかしなくてはならないのが少し面倒ですね。また，外部関数の中にその外部関数のアドレスを返す関数を入れておき，X-BASIC上で実行するという手もあります。多少手間はかかりますが，一度作っておけば便利かもしれません。

アドレスさえわかれば，あとはなんとかなりますので，がんばってみてください。

（影山 裕昭）

質問にお答えします

日ごろ疑問に思っていること，どんなことでも結構です。どんどんお便りください。難問，奇問，編集室が総力を上げてお答えいたします。ただし，お寄せいただいているものの中には，マニュアルを読めばすぐに回答が得られるようなものも多々あります。最低限，マニュアルは熟読しておきましょう。質問はなるべく具体的に機種名，システム構成，必要なら図も入れてこと細かに書いてください。また，返信用切手同封の質問をよく受けますが，原則として，質問には本誌上でお答えすることになっていますのでご了承ください。なお，質問の内容について，直接問い合わせることもありますので，電話番号も明記してくださいね。

宛先：〒108 東京都港区高輪2-19-13
NS高輪ビル
ソフトバンク株式会社出版部
「Oh! X質問箱」係

FROM READERS TO THE EDITOR

まだまだ暑い日が続きますが，いよいよ食欲の秋！　暑さで失った体力を取り戻すため，おいしいものを探してしっかり食事をしましょう。体重計はしばらく押し入れにでも隠しといて，秋の味覚を堪能しようじゃありませんか。

◆注文したX68000XVIが届かないので，X-BASICについてはまだわからないのですが7月号の特集を読んだ感じでは少々てこずりそうです。COBOLなら仕事で使っているため自信がありますけどね（COBOLは簡単だからなあ）。

花野　信行(27)千葉県

細かい仕様は違ってもプログラミングの基本は同じです。ある言語を修得しているなら，とっかかりさえつかめれば大丈夫ですよ。

◆7月号の付録を見て，以前にPC-6001mkIIを使っていた頃を思い出しました。PC-6001mkIIにもこの付録と似たような小型のリファレンスが付属していて，本体を田舎から譲り受けるまでの1年間ほどは，そのリファレンスのコピーを片手にピコピコゲームの思索にふけっていたものでした。この付録のおかげでX-BASICが使えるようになる人はきっとたくさん出てくると思いますよ。

寺門　修司(19)兵庫県

◆7月号付録の「X-BASIC REFERENCE BOOK」は便利ですね。ちょっと機能を確認したいとき，わざわざ大きなマニュアルを見る必要もないし，手軽に使えます。こういった付録は大歓迎です。

鈴木　明敏(17)長野県

具体的な仕様はマニュアルを参考するなりして，使い分けながら有効に活用してください。

◆「Small-C」のディスクどうもありがとうございました。電脳倶楽部の「滋養強壮」のディスクラベルを使っていましたが，カゼで体調を崩している私にはぴったりのプレゼントでした。現在は「ついにCができる」という感激と「また不眠不休の日々が続く」という不安でいっぱいです。ところで，大阪近辺在住の方は知っていると思いますが六甲山といえばハイキングに最適の山です。しかし，ベテランの登山愛好家も暇を見つけては六甲山に行きます。「六甲山に始まり六甲山に終わる」だそうです。私のパソコンライフもBASICに始まりました。そして，BASICに終わるのかもしれません。BASICも基本的なものだけど奥が深いようです。

深川　哲光(32)香川県

要は使い方しだい，といったところでしょうか。皆さんもいろいろな使い方を見つけていってください。

◆支払いの達人の話を読んで私もさっそく実行してみようと思い，スーパーへ行って198円の牛乳をレジの前に持っていきました。そして，おもむろに財布から198円を取り出そうとしたときレジの人が言いました。「203円になります。」The Master of Paymentには，まだまだなれないようです。

佐渡　詩郎(15)石川県

消費税というドラゴンにあえなく玉砕してしまいましたか。もっとレベルを積んで再度挑戦しましょう。

◆The Master of Payment，いやあいるんですねこういう人が。支払いの達人，こんな人が増えれば小銭がそこいら辺に散らばることがなくなるでしょう。それにしても自分の財布の中には5円玉の嵐。思いのほか5円が半端になるんです。

寺内　章(24)埼玉県

細かいお金ほどうっとおしいものはないですね。一度でいいから「いやあ，福沢さんが余ってしょうがないよ」と言ってみたい。

◆ショック！ ハードディスクのFATが破壊され，せっかく作ったSXIMAGE高速版がソースリストごと消えてしまった。ファイルを放り込んでから15秒で表示するくらいまで速くしたのに……。破壊されたのは，電源OFF割り込みでテキスト画面を待避するプログラムを作っている途中でした。くやしいことにこっちのほうは全部残ってます。一応動きますし……，あ～くやしい！また作り直して送るからほかの人のを載せちゃいやだよ。

西田　文彦(20)神奈川県

不幸は突然やってきます。バックアップはこまめにね。

◆平成5年度から中学の技術家庭科にコンピュータが取り入れられるようですね。私の母校（三国中学校）も校舎の一部を改築してコンピュータ室（端末は20台，テレビで見たがX68000ではなかった）を作ったそうです。あと20年遅く生まれたかった，と思ったけど自分のときとは比べものにならないほど受験戦争も激しくなっていただろうし，人生はうまくいかないなあ。

岡部　誠(26)福井県

うらやましがるのもいいですが，ひょっとしたら学生時代から強制的にコンピュータを触らせられたおかげで，コンピュータが嫌いになっていたかもしれませんよ。

◆卒業研究のためNEWSのX-WINDOW上で3Dグラフィックのアニメーションをやっています。実際に自分たちでDōGAのようなものを作ろうとしてみてわかりましたがすごく難しいですね。ところで宮城N○Cに内定が決まりました。面接で，「持っているパソコンはN○C製？」と聞かれて「はい」と答えてしまいました。まあ，内定したらこっちのものですよね？

昆　信(21)北海道

う～ん，弱いなあ。もっと胸を張って「私はシャープユーザーだ！」と言えるようになりましょう。

◆X68000XVIの発売される1カ月前にX68000 SUPERを買った人もかわいそうだと思うが，SX-WINDOWのVer.1.1が発売されたことを知らずにVer.1.0を買ってしまった僕も悲惨だと思う。関係ないけど常駐プログラムを初めて作ろうとしたんですが「CAPSキーが1秒ごとに点滅する

◀岡村　直也　兵庫県
最近投稿されるイラストはおとなしいので君の4コマは光っているぞ。う～ん，7月号からこういう展開になるとは……3分間笑い転げちゃったい。

◀住友　智代　愛知県
もしかすると怪物の肉も結構おいしかったりして。一度は食べてみたいなあ，青い血のしたたるもも肉を……。

プログラム」が「1秒ごとにバスエラー，アドレスエラーを発生させるプログラム」になってしまいました（律儀に繰り返すんだよなあ）。

稲富　顕二(19)岡山県

今度は目を皿のようにして情報を見逃さないようにね。

◆シャープパソコンフォーラム'91でのことだった。ウルフのブースにあったソルフィースをキーボードでプレイしていたが最後でしくじってGCS-WTの手前でGAME OVERになってしまった……。が，後ろで見ていたお兄さん方，頼むからその「変なもの」を見るような目つきだけはやめてくれ～。おれは人間だ～！ ちなみに現在キーボードでクリアできるものは，グラディウス，ソルフィース，ドラゴンスピリット，A-JAX，ファランクス，パロディウスだ！，沙羅曼蛇，メタルサイト，こんなもんかな。

小林　信裕(18)千葉県

僕がキーボードでクリアしたゲームは，ちゃっくんぽっぷぐらいだなあ。それにしてもすごい。

◆8月号から定価600円ということでこれからの値上げ予想だ！ 480円から540円までが64カ月，540円から600円が51カ月で上げ幅はともに60円。このデータから次は40カ月後の155号の定価は660円となり，その次は32カ月後の187号で720円になるんですね。で，恐ろしいことにこのペースだと314号の34回目の値上げからは1日ごとに定価が上がり，1万円を突破するのも同じ314号なのです(笑)。ああっ，Oh!Xの読者に未来はあるのだろうか（ウソ度100%）？

斉藤　修(23)宮城県

ああ，読者の財布に明日はあるのか！

◆「サイレントメビウス」をやっと買いました。いわゆる「ゲームファン」の人からはさんざんいわれそうなゲームではありますが（簡単すぎる，というのがいちばん多そう）アドベンチャーゲームとしてはオーソドックスな作りだし，操作性も悪くなかったんで及第点をあげられるゲームだと思っています。なんといってもひたすら「見せる」ことに徹した制作姿勢がいいですね。すでにゲームではないという気がしますが。システムサコムが「ノベルウェア」ならガイナックスは「コミックウェア」というところでしょうか。

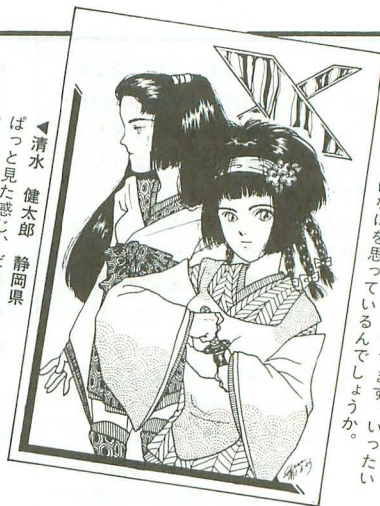
西島　郁夫(23)大阪府

原作の雰囲気をそのまま生かした作りはなかなかですね。しかし，値段と内容のつりあいが問題かな。次回作に期待しましょう。

◆最近久しぶりにX68000の電源を入れた。X-BASICでゲームを作ろうと思いプログラムを打ち込んでいると不思議なことに1回もバグが出なかった。こんなことは初めてだ，と調子にのってセーブをしないでいたら突然ブレーカーが落ちた。原因はクーラーの使いすぎである。そういえば去年も同じようなことがあったなあ。とても悲しい。

宇野　寛和(15)東京都

夏になるとよくあることとはいえ経験したくないものです。電気はこまめに消そうね（あたしゃ東京○力かいな）。

◀清水　健太郎　静岡県
ぱっと見た感じ，だまし絵みたいでおもしろいね。でたらめなんていってるけど，そういうことにとらわれずに自由な発想で描いていいと思うよ。

◀新井　美香　北海道
ずいぶんきれいにまとまって見やすいイラスト。女性らしい繊細な感じがよくでてます。いったい彼女たちはなにを思っているんでしょうか。

◆ついにエアコンを買った！ 今年の夏は暑さを忘れることができそうだ。しかし，自分の財布を見たら寒気を感じた。う～ん，暑さを忘れるどころか今年の夏は寒くなりそうだ。ウ～ブルブル……。

菊田　俊彦(32)茨城県

エアコンの使いすぎで電気料金が跳ね上がり，支払いが不可能にならないように祈っています。

◆そろそろX68000にもCD-ROMがあるといいな，と思い始めた今日この頃。MOはちっとも普及しないし，いつまでもフロッピーベースのソフト供給は情けない。今度発売されるメガドライブ用のCD-ROMがうらやましいです。ゲームをするだけならX68000よりもメガドライブ＋CD-ROMのほうがいいでしょう。それから1024×1024表示できるディスプレイがあればSX-WINDOWももっと使いやすくなるんですけどね。最近はそういった楽しいオプションが出ませんね。シャープに催促する意味でオプション特集でもやりましょうよ。

桜井　良太郎(20)東京都

具体的になにがほしいのか，皆さんの考えを聞きたいな。

◆メガドライブ用のCD-ROMはすごい。回転拡大縮小，PCM8音，メガドライブより高速な12.5MHz68000MPUなど。このような本体の性能を大幅にアップさせる周辺機器がX68000にも出ないだろうか。そうなるとますます，X68000はゲーム機だ，といわれそうだが。

植木　隆史(21)千葉県

目的をゲームに限らなければ，そういう誤解を招くことがなくなるんじゃないかな。

◆今年は1カ月に1本ゲームを買っている（珍しい）。どんなに疲れていてもちゃっかり買いにいって夜遅くまで遊んでいるから，最近人から「顔が青い」と言われます。確かアニメで青い顔って悪者の象徴でしたよね，う～ん。

菅谷　英明(24)兵庫県

じゃあ，菅谷さんは悪の大魔王ってところですか。

◆「対談GMコンポーザー」は非常にナイスな企画ですね。不定期でもいいから続けてください。古代祐三，永田英哉，磯田健一郎，小沢純子，めがてん，かわげん，中潟憲雄，大野木宣幸，などなど。ネタはつきないと思いますのでぜひやりましょう。もちろん完全作品リスト，プロフィールもよろしくお願いしますね。

高橋　雅(19)福島県

さあて，次は誰がくるか楽しみにしていてください。

◆6月は残業が多かった。当然手当は多い。そして，6月はボーナス……ウヒョ！ なにを買おうかな。まずは3年間3着しかなかったスーツ（少し恥ずかしい）にオサラバしてスーツを買おう。残った金でX68000XVIでも買おうかな。と思ってスーツを買いにいったがスーツのほかにもシャツやネクタイ，ネクタイピンに靴……エエッもう自由に使おうと思っていた金がこんだけ!? そうか，パソコンと同じくスーツ本体だけじゃなく周辺機器がいっぱい必要なのね。

森崎　高重(21)愛知県

芋づる式に出費がかさんでしまいましたか。今度はX68000の周辺機器を充実してあげましょうよ。

◆この間，学校の実習で溶接をやりました。アーク溶接はすごく楽だったけどガス溶接はすごかった。みんな爆発しまくりですごく怖かった。ガス溶接は二度とやりたくない。

山下　洋(16)大阪府

いかん，そんなことでは立派な溶接工になれないぞ。でも，ケガ人が出なくてよかったね。

◆6月18日，私は「食品衛生学」の講座を聞きつつ「Oh!X」を読むというマルチタスクをこなしていた。しかし，この作業は「満開の電子ちゃん」のおかげで非常に苦しかったことはいうまでもない。祝さ～ん，あの無理やりな展開が好きです。

石崎　賢(21)千葉県

で，8月号の「満開の電子ちゃん」はどんな講座を聞きながら読んでいたんですか？

◆朝日や夕陽が強烈に差し込む日に電車に乗ります。もし，あなたの乗ったその電車の窓の片側から日光が差すようならその日光と向かい合うように座りましょう。日光が建物や電柱に遮られ電車のスピードにリズムを刻みます。そしてあなたは目を閉じます。光がまぶたの裏で踊り電車の揺れが遺伝子の揺らぎとシンクロした

とき……。　　　　　　　　　　橋本　忍(20)埼玉県

いきなりシンクロエナジャイザーネタのハガキ。でも，電車の中でそんなことをしてトリップしたまま車庫に入ってしまったら，ただのバカですね。

◆アンケートハガキにミシン目が入ってなかったのは私のOh!Xだけなのでしょうか？
　　　　　　　　　　　　　　陣山　達夫(21)大阪府

7月号のハガキの中でいちばん多かったネタ。こちらの手違いです。ごめんなさい。

◆本当にバイトをしたら金使いが荒くなるんですね。僕なんかいつもお金があるのをいいことに欲しいものがあるとすぐに買ってしまうようになりました（といっても安物，高いものはすぐに買えん)。　　　　野口　博俊(15)福岡県

そこはがまんのしどころ。しばらくがまんして，今度はすかっと大きな買い物をしましょう。気持ちいいですよ～。

◆急にアドベンチャーゲームがやりたくなって「黄金の羅針盤」を買いました。まだ終わっていませんがなかなか趣がいいです。曲もいい。でもグラフィックがほとんどPC-9801と変わらないのが残念。私はタイルパターンが嫌いなのです(PC-9801ユーザーの友人に，贅沢いうんじゃねえ，と怒られました)。なんかざらざらしているみたいでいやだな。話は変わりますがX68000XVIが発売されたというのにちっともテレビCMがないですね。う～ん，ユーザー任せすぎですよ，へぐへぐ。栄作ちゃんは熱すぎるのでヤメにして誰か強力なイメージキャラを使って広告してほしいです。個人的にはラビー（関根勤）がよいけど……ダ，ダメですね（笑)。
　　　　　　　　　　　　　鈴川　美佳子(18)東京都

取り込み画像の関根勤がでで～ん！ と登場するようなCMはちょっと遠慮したいな。

◆先日，書店に本を買いにいった帰り見知らぬ高校生に呼び止められ「お前，どこの中学だ！」と言われてしまった。そんなに幼い顔でもないと思うんだけどなあ。ちなみにそのときは驚きのあまり「俺は高校生……」と口走ってしまった。本当は大学生です。鈴木　賢吾(20)北海道

もう少し若く見られたら子供料金でいろいろできて得をしたかもしれませんね。

◆6月30日の夜10：30頃，私はX68000版のグラディウスに夢中になっていた。4面を終わって5面に入った直後のことである。いきなり私の家で飼っている猫のマイケルが，私の握っているジョイスティックの横に置いてあるキーボードの上を歩こうとしたのでESCキーでゲームを止めてマイケルを降ろそうとしたが先にマイケルがブレイクキーを押してしまったのである。しかたなく小室哲也のソロアルバム「Digitalian eating breakfast」を聞きながらこのハガキを書くことにした。マイケル，じゃれていないで俺にグラディウスをやらせろ。皆さんゲーム中は猫を近づけないようにしましょう。
　　　　　　　　　　　　　　太田　哲夫(19)神奈川県

でも，やっぱりかわいいんでしょう？

◆「誰にでも身に覚えがある勘違いシリーズ(その1)」「月極駐車場」という看板を見て「へえ，この月極っていう駐車場チェーンは全国にあるのか」と本気で思ったこと，あなたありませんか？　　　　　　　　　荒木　芳典(21)岡山県

ありません。

◆今日，ひとり2匹の割り当てでミミズを解剖した。39～40°のお湯（微温水）に入れて4～5分間放置すると動かなくなる。これがミミズの麻酔だという。外部形態のスケッチのあといよいよハサミを入れる。消化管や背行血管を傷つけないように切り開くのはなかなか大変だった。実験が終わって帰りに寄ったコンビニで友人がひと言，「ヤキソバが食べたい」。生物科の明日はどうなる？　　　　　岩瀬　貴代美(19)福岡県

まあ，ミミズぐらいならいいですが人体解剖のあとに「焼きうどんが食いたい」なんて言う強者がいたら……やだなあ。

◆最近ページの端で手を切ってしまうというミスを毎月，必ずやってしまいます。同じミスを繰り返しやるということは早くも痴呆症が始まっているのでしょうか。　　上居　忍(20)北海道

毎月，読者ハガキで指を切ってしまう僕ですがそんなときは「これは我々に対する挑戦だな」とごまかしてしまいます。

◆7月号のmicroOdysseyを読んでなるほどと思いました。確かにパソコンのソフトはあまり値段が下がりません。安く買うためには中古ソフトなどを買うか，お店にまけてもらうしかありません。しかし，まけてもらうといっても値引きはあまり期待できないし。これからはもっと安くなってほしいものです。なにしろお金がもちません。　　　　　　　　永井　徹(18)滋賀県

安いことはいいことですが，値段が安くなったら作品の質が落ちてしまった。なんてことになったらしゃれにならないしね。

◆7月号の付録「X-BASIC REFERENCE BOOK」はなかなかよい。以前，自分で書こうとしたがあまりに大変なため途中であきらめてしまったが，こうしてめぐり会えるとは，ひょっとして私とX68000は赤いOh!Xで結ばれているのかもしれません。ちなみに7月号には「X68000」の文字が約328個あります。　　　志賀　宗一(17)愛知県

今度はしっかり個数を数え，統計をとってみましょう。出てきたデータが役に立つとは思えないけど。

◆X68000ACE-HDを修理に出したらシャープの倉庫が火事で水浸しになったようです。そのため新しいX68000ACE-HDが帰ってきたわけですがこいつのVRAMの調子が悪く，またまた交換してくれることになりました。しかし，もうX68000ACE-HDの在庫がなくてX68000EXPERT-HDが帰ってきた。やった，20Mバイト分のハードディスクを得したぜ。でも色が黒しかないんだって。モニターはグレーなのにさ。やっぱりX68000はオフィスグレーだと思いませんか。オフィスグレーを愛してやまない男より。
　　　　　　　　　　　　　　太田　貴道(18)奈良県

やっぱり「黒の統一美」ですよ（ちょっと恥ずかしい)。

◆メガドライブのアドバンスト大戦略をやっていますがこれがまたすごく面白い。X68000にも，これくらいパワーのあるソフトがどんどん発売されてくれればいいと思う今日この頃です。
　　　　　　　　　　　　　　相沢　孝之(20)栃木県

また，なにか面白いゲームが発見できたら教えてくださいね。

◆一生懸命バイトをして手に入れたCM-64がいまいち有効に使われていない気がする。Mu-1もバージョンアップしたようだし，そろそろ音楽の勉強を始めようかな。話は変わるけど電波新聞社から発売される「イース」のデモを見た。あまりの絵の細かさに驚いてしまった。こりゃあ買うしかない。でも財布の中身が心配だ。
　　　　　　　　　　　　　　小沢　健一(17)静岡県

原作とは違った趣で，電波新聞社らしいこだわりの「イース」。どれだけユーザーの評価を得られるか注目したいですね。

◆最近，X68000の前でボケ～とすることが多くなってしまった。プログラミングするわけでもなくゲームもすぐつまらなくなってしまう。これはまさか破局の前触れ「倦怠期」がやってきたということではないだろうか。なんとかして交流を取り戻さなくては，などと感じています。これはスランプとでもいうのでしょうか。皆さんはこのような経験はありませんか？
　　　　　　　　　　　　　　堀尾　忠教(18)鳥取県

▼薮田　俊平　和歌山県
仲間がいないなら作ってしまおう，の精神で周りにいるPC-9801ユーザーを引きずり込みましょうよ。電子ちゃんみたいにね。

▼加藤　敬志　福岡県
おやま，ダルジイがずいぶんかわいいなあ。ひょっとしたら，ゲームの外ではこんな日常的な光景が見られるのかもしれませんね。

気ばかりあせってもしょうがないですよ。のんびりアイデアでも考えていれば，そのうちやることが見つかりますって。

◆暑い季節がやってきた。暑さが苦手の私にとってとてもつらい。ましてや夏休み中にT定規を使ってエンジンの製図なんかやってられない。早く実家（北海道）に一時疎開したい。
神生　直敏(21)栃木県

いいですね，疎開する場所があって。僕もどこか会社以外で涼しいところに行きたい。くう～。

◆7月7日から大相撲夏場所が始まります。私の家から10分ぐらいの寺に藤島部屋が毎年くるんですよ。去年は貴花田とアクシュしてしまいました（予想よりもはるかにでかくて硬い手でした）。さて，学生にとっていちばん楽しいとき，そう夏休みがもうすぐ始まります。今年はX68000でマシン語の勉強をする予定です。もちろんXC Ver.2.0をバイトして買うつもりです。とっても楽しみ。でもでもでも～，今は期末テスト9日前だあ！　加藤　恵吾(15)愛知県

今はもう楽しい夏休みですね。目標は達成できましたか。ところで，課題，レポートは片づいているのかな？

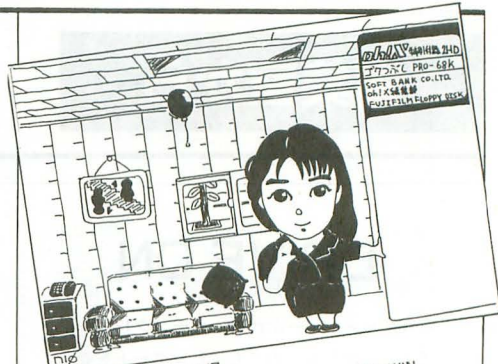
◆プログラミングをしているときにわからないことがあると，よくOh!Xの質問箱のコーナーを読みます。質問箱で取り上げられた質問を読んでいると自分が知りたかったことが結構あります。でも，いざ探そうとなると過去2年半分のOh!Xをかたっぱしから探すのはちょっと大変。そこでCBASE.Xを使って質問箱のデータを作って引き出せるようにしようと考えています。ところで，あの謎の姉弟4コマ漫画で有名な岡村直也さんは，いったいおいくつなんでしょうか？　まさか，ピンポンパン体操を知っているとは……。
本田　英雄(21)埼玉県

う～ん，元ネタがわかる本田君が21歳なんだからたいして違わない気がしますけど。

◆6月から東京ぐらし。新宿で寝起きして新宿に歩いて出勤する，楽な生活ではありますがワンルームのマンションの家賃は給料手取りの半分ほど。おまけにX68000XVIまで買ってしまったし。ついでにXC Ver.2.0も買いましたがなるべくゲームは買わないぞ，と。野田　直人(33)東京都

なるべく買わない，というのが正直でいいですね。それにしても給料の半分の家賃とはぶっとんでいるなあ。

◆最近どうにかC言語の山を越えることができて，ようやくあの高いコンパイラを有効に活用できるというものです。友達が言うには，「XCのマニュアルは親切で，ほかのよりもわかりやすいマニュアルだ」そうです。でも僕は1年かかった。まあ，これからも山あり谷ありだろうけど必ずものにするぞ。　鬼谷　秀樹(19)福岡県

今はちょうど夏休み。時間はたっぷりあるでしょうからがんばってね。

▲大村　直人　岩手県
なんのイラストかなあ，と思っていたらSX-WINDOWですか。オープンされたウィンドウがあればもっとどんなイラストかはっきりしたと思うよ。

ぼくらの掲示板

●掲載ご希望の方は，官製ハガキに項目（売る・買う・氏名・年齢・連絡方法……）を明記してお申し込みください。
●ソフトの売買，交換については，いっさい掲載できません。
●取り引きについては当編集部では責任を負いかねます。
●応募者多数の場合，掲載できない場合もあります。
●紹介を希望されるサークルは必ず会誌の見本を送ってください。

仲間

★「PC-EMPIRE」ではサークル発足にともない新規会員を募集します。会員の投稿をメインにした毎月1回発行の会報と月1回のゲーム大会をメインに活動しています。対象機種はX68000，FM TOWNS，PCシリーズなどです。興味を持った人は62円切手を同封のうえ，下記の住所へ連絡してください。折り返し入会案内書を送ります。〒716　岡山県高梁市津川町959-1　渡辺　勇(17)

売ります

★X68000，X1専用スキャナ「CZ-8NS1」を75,000円で。X1，X1turbo用カラーイメージボード「CZ-8BV1」を8,000～10,000円で。X1，X1turbo用立体映像ボード「CZ-8BR1」を8,000～10,000円で。いずれも箱，マニュアル完全にあり。連絡は往復ハガキにて。〒699-55　島根県鹿足郡六日市有飯232-2　落合　康一(31)

★X1用FM音源ボード「CZ-8BS1」を送料込み10,000円で。付属品，マニュアルすべてあり。使用期間は2年間。連絡はハガキでお願いします。〒800-02　福岡県北九州市小倉南区葛原2-14-23　恒松　裕朋(17)

★X68000用拡張I/Oボックス「CZ-6EB1」＋増設RAMボード「PIO-6BE2（2Mバイト）」を送料込み60,000円で売ります。別売りも可。半年間使用，完動品です。〒308　茨城県下館市小川1385-7　鯨　雅之(34)

★熱転写カラー漢字プリンタ「CZ-8PC3」を25,000円で売ります。ケーブル，説明書あり。動作は完全に保証します。連絡は必ず往復ハガキでお願いします。〒921　石川県金沢市窪1-55-1　中川　一実(42)

★X68000用カラーディスプレイ「CZ-606D（グレイ）」を40,000円で売ります。保証書，箱，説明書あり。連絡は往復ハガキでお願いします。〒989-23　宮城県亘理郡亘理町字中町東219-14　佐藤　哲也(24)

★X68000ACE，PRO用内蔵増設RAMボード（1Mバイト）「PIO-6BE1-A（IOデータ製）」を送料込み16,000円で売ります。保証書，箱，説明書，付属品すべて新品です。連絡は往復ハガキで。〒981-33　宮城県黒川郡大和町小野字御仮屋19　平野　智幸(21)

★COMSTARZ　2424/5を送料込み20,000円で。保証書，箱，説明書あり。連絡は往復ハガキでお願いします。〒704　岡山県岡山市西大寺浜150-2　吉川　貢司(17)

買います

★Rolandの「CM-64」を6～7万円で。できれば箱，マニュアルありで少々の傷は可。安くしてくれる方にはなにかオマケをつけます。連絡は往復ハガキで考えさせてもらいます。〒389-08　長野県更級郡上山田町温泉2-25-7　山崎　高志(17)

バックナンバー

★Oh!Xの1990年7月号，10月号，11月号を各600円で譲ってください。切り抜きは不可。できたら600円ですがどうしてもという場合には1,000円まで考えさせてもらいます。連絡はハガキでお願いします。〒921　石川県金沢市窪1-81　新井由之(14)

★Oh!Xの1990年3月号，5月号を各1,000円（送料込み）で買います。連絡は住所，氏名，年齢，電話番号を明記のうえ必ず封書でお願いします。〒667　兵庫県養父郡八鹿町岩崎438　藤坂　潤

★Oh!MZの1986年9月号を3,000円で，1987年10月号を1，500円で買います。いずれも送料込みです。切り抜きなどある方はどんな状態か明記してください。連絡は必ず往復ハガキでお願いします。〒164　東京都中野区中野5-42-8　栗原景(19)

編集室から from E・D・I・T・O・R

DRIVE ON

今月からメンバーが代わったモニタレポートです。なかなか力のこもったレポートを新しいモニタの皆さんからいただきました。

●特集「Personal Tool, BASIC」の冒頭,「パソコンでプログラミングをする」と題する部分には,私も強い共感をおぼえました。実際,プログラミングで大切なのは,プログラミングの技術ではなく,「それ以外の」ものなのだということを最近あらためて実感しているのです。私はC言語を始めようとしてC compiler PRO-68Kを買ったのですが,ほとんど使っていません。なぜかと考えてみて,作りたいプログラムがないからではないか,と思いました。いまだにきっかけをつかんで「初めの一歩」を踏み出すことができないのです。今回の特集はいいきっかけになる気がします。

宍戸 輝光(17) X68000 PRO 東京都

●私がアセンブラを始めた理由は「コンピュータとは何だ?」という疑問を解決したかったからであり,決してコンピュータを自分の手や足となる「道具」にしようと思ったからではありません。「A」という1文字を表示させて得たものはただひとつ,「コンピュータはバカだ」ということだけでした。逆に,「道具」にしようと考えるのであれば,やはりBASICがいちばんお手軽なのでしょう。この「道具」というのは,計算機として,文章作成機として,はては「パーソナルコンピューティング」という言葉につながっていくとしたら,そのインタフェイスはやはりお手軽,便利なBASIC。遅くたっていいじゃない。BASICって強力ですよ。つくづく,そう思いました。

安井 百合江(17) X68000 PRO 愛知県

●私がBASICを使うのは主に音楽だけですが,昔のPC-6001mkIIと比べると,とても使いやすいのは事実です。あと,たまにXIの曲も打ち込むんですが,音色を移すのが大変で,一度XI→X68000音色移動ツールを作ってみようかと考えたこともあります。ですから今回の特集では,BASICを使いこなせればたいていのことはできるんだな,というのがわかったことがいちばん自分としては自信づけられました。せめて,自分がほしいなと思ったものだけでも自分で作ることができれば,それだけで十分だと思います。でも,初心者にはまだX-BASICの壁は厚い気がします。

谷口 有香(22) X68000, PC-6001mkII 北海道

●6月号で「X-BASICポケットリファレンスブック」の企画を知ったときには,はっきりいってがっくりしました。なぜかというと,X-BASICでは,その高級言語さゆえに,付属のBASICマニュアルさえあれば事足りると思ったからです。しかし,実物を見たとき考えが変わりました。これははっきりいって使えます。何かあるたびにあの分厚いマニュアルを引っ張り出さなくていいのですから。

安岡 毅(17) X68000 PRO, MSX 京都府

●「パロディウスだ!」のレビューで,6月号では西川善司さんが「マニュアルだよーん」といい,7月号では中森章さんが「オートでなくっちゃ」といっていました。ライターが互いに違う意見をいえるっていうのはいいのではないかと思います。読者は混乱してしまうかもしれませんが,それだけ読者が自分の意見を見つけ出せていいと思います。

弦元 達也(20) X68000 ACE 香川県

●私は純粋な(?)XIユーザーですので,数少ないXI関係の記事をレポートさせていただきます。さて,「THE MASTER OF PAYMENT」。表紙にもありましたし,側面にも「XI/turbo用買い物ゲーム」とあったので,「ミュール」(懐かしい)に似たようなものかなと思ったら,全然違いました。私の財布も小銭は少ないですが,これは別に純粋理性的だからではなく,単なる金欠によるものです。しかし,たまに小銭がいっぱいあると,目的の硬貨を取り出すのが困難です。そうすると,買い物の際にいかに硬貨を効果的に(英訳したら通じないギャグ)減らすか,ということになります。で,ためしに打ち込んでやってみたのですが,ファーストプレイのときにレベル2で終わった私はやっぱりフツーの計算力もないんでしょうね。でも,展開もスムーズで一風変わった面白いゲームでした。これからもがんばってください。

中村 圭介(16) XIG 神奈川県

●「善バビ番外編 対談・GMコンポーザー」について。日本のゲーム業界は作成者をおろそかにする傾向があり,なかなか前面には出てきません。こういうコーナーができることで,作成者が注目を浴びればやる気も出るし,話を聞くことはこういう職業を目指す人の参考にもなり,いいのではないでしょうか。

功刀 和久(22) X68000 ACE-HD, XIturbo 埼玉県

●私も「こんなツールを作りたい」と思うことがある。が,作れないのである。FOR文がループ,IF文が条件などとは知っているのだけれども,それを使ってプログラムを組むことができないのである。単語,文法を知っているのだけれども,英文が書けないのである。私が初めてBASICで組んだのは,題して「width96の画面いっぱいにl(エル)を書いて,そのなかにひとつだけある1(いち)を探す」というものだ。もちろんRUNするたびに1の場所は変わる。当時はこんなものでも結構頭を使った。

いまでも私はツールが作りたくても思うようにいかない。だから,人のプログラムを見て,「ああ,こうしたいときはこうするのか」と学んでいかなくてはならない。今回の特集ではBASICの奥の深さを見たように思われる。「BASICは何でもできるんだ」というのは間違いではないと思う。

遠藤 隆一(17) X68000 PROII, XIturbo 北海道

ごめんなさいのコーナー

8月号 基礎からのカラー印刷

P.94 リスト2に誤りがありました。

530 for z=0 to 2

と訂正してください。

8月号 吾輩はX68000である

P.62 本文中の記述に誤りがありました。

"……それぞれwhile文,repeat文と呼ばれている"というのは,"while文,do文"の間違いです。

8月号 IOCS用FONT200書体

P.105 製品の価格が間違っていました。正しい値段は3,000円(税込)です。たいへんご迷惑をおかけしました。

バグに関するお問い合わせは
☎03(5488)1311(直通)
月～金曜日16:00～18:00

お問い合わせは原則として,本誌のバグ情報のみに限らせていただきます。入力法,操作法などはマニュアルをよくお読みください。

また,よくアドベンチャーゲームの解答を求めるお電話をいただきますが,本誌ではいっさいお答えできません。ご了承ください。

MAGICはもはや不思議ではない

▼5月号でその姿を現し，さまざまな反響を呼んだX68000版「MAGIC」。実際に使ってはいない人でも，「SION」をプレイして「MAGIC」に興味を持っていただいたことと思います。発表後，読者の皆さんからたくさんの投稿をいただき，さらに「MAGIC」はパワーアップしつつあります。3D画面の魅力は皆さん感じてはいるでしょうが，自分で作るとなるとなかなかにむずかしいものがあります。それをサポートする，この「MAGIC」。さらなる飛躍を誰もが待ち望んでいることでしょう。

▼今月から新しくなったモニタレポートのメンバー。内容は左のページを見ていただくことにして，面白いものが送られてきたので紹介しましょう。野原志貴乃さんという方がマウスのぬいぐるみ（？）を送ってくれました。「突然，変なものを送ります。ほかのモニタさんのようにむずかしいことが書けないので，ワイロです。……ほかにディスク型のざぶとんとディスクのぬいぐるみを作ったのですが，ざぶとんは主人のお尻の下，ぬいぐるみは娘の腕の中にあります。マウスのぬいぐるみは主人や友人からリクエストがあり，たくさん作ったのでおすそわけです。……では，またなにか思いついて作ったら送ります」

ということです。ぬいぐるみの出来はなかなかのもので，表に出すと一瞬にしてなくなりそうなので，大事に保管してあります。また，レポートのほうですが，むずかしいことなんか書かなくていいですから，思ったことをどんどん文章にしてください。ほかのモニタの皆さんも，ね。

▼突然，特別企画ということでお送りした「Oh!Xの正しい読み方」。新しい読者の皆さんには道標として，そして以前からの読者の皆さんにはOh!Xのたどってきた道程を再起させるものとして役立つことでしょう。Oh!Xがさらに楽しくなることは間違いありません？

▼作者多忙のため，「よいこのためのSX-WINDOW講座」と「知能機械概論」はお休みになってしまいました。申し訳ありません。

投稿応募要領

- 原稿には，住所・氏名・年齢・職業・連絡先電話番号・機種・使用言語・必要な周辺機器・マイコン歴を明記してください。
- プログラムを投稿される方は，詳しい内容の説明，利用法，できればフローチャート，変数表，メモリマップ（マシン語の場合）に，参考文献を明記し，プログラムをセーブしたテープ（ディスケット）を添えてお送りください。また，掲載にあたっては，編集上の都合により加筆修正させていただくことがありますのでご了承ください。
- ハードの製作などを投稿される方は，詳しい内容の説明のほかに回路図，部品表，できれば実体配線図も添えてください。編集室で検討のうえ，製作したハードが必要な場合はご連絡いたします。
- 投稿者のモラルとして，他誌との二重投稿，他機種用プログラムを単に移植したものは固くお断りいたします。

あて先

〒108 東京都港区高輪2-19-13 NS高輪ビル
ソフトバンク出版部
Oh!X「㋢㋴㋮㋱」係

SHIFT・BREAK

▶編集部の環境で仕事しながらこう考えた。DIを嫌いだというと角が立つ。主張をすると聞き流される。2Mしかないマシンは窮屈だ。とかくX68000は奥が深い……。たしかに私と編集部の環境にギャップがあるのは問題だ。しかたなく自分のシステムディスクで起動するのだが，今度はSCSIが使えなくなるので結局ダメなのは変わらないのである。（八）

▶就職試験に落ちてしまった。会社を訪問したときは，「絶対に来てくださいね」といっていたのに，いきなり落とすんだもんなあ。次の職を探さなきゃ。話は変わるが，先日Oh!MZのバックナンバーを読んでたら(浦)君の名前を年間モニタのなかから見つけた。そういえば読者のハガキのなかに(で)君もいたし……。みんな読者だったんだね。（毛）

▶で，毛内氏のいうとおり，私はモニタの出身なのです。技術がなくてもOh!Xのスタッフにはなれるという，いまのところ唯一の見本みたいなもんです。「Oh!Xのスタッフになりたい！」という人，プログラムに自信がなければモニタの道から入るのもいいかもしれないよ。しかし，モニタ出身者の目から見ても，安井百合江嬢はすごいなあ。（浦）

▶大学に行くために車を使おうと思ったら右リアタイヤがエア洩れしていた。給油ついでにエアを入れてもらおうと思ったが，あいにくガソリンは満タン。しかたなく大学の通り道にある○○石油に寄ったらエアだけではだめだと。ところが，その先にあるゼネラル石油では無料でエアを入れてくれる親切ぶり。利益を無視したサービスに感動した。（H.K.）

▶ひとまず，ホッとしている。しかし，これからも多くのことを学ばなければならない。同じところに居続けるのはラクだけれども，辛くても前へ前へと進み続けようとすることが必要なのだ。おかげで忙しい毎日ではあるが，また近いうちにこういった機会を持てたらと思っている。それにつけてもA氏にはただただ感謝。（やはり次はchaosのC）

▶ともすれば，寄生獣は90年代のデビルマン(原作)である。悪魔と合体したが，心も身体も8割は人間のデビルマンと，寄生獣と合体したが，心も身体も8割は人間の「寄生獣」。悪魔の存在を隠すデビルマンと寄生獣の存在を隠す「寄生獣」。デビルマンに流れるロマン（シレーヌは美しい）を排除した点が'90年代だ。名作になる可能性大である。（K）

▶アメリカに留学している知人から，fastdhry.cなるプログラムをもらった。これはドライストンベンチマークを，C言語のソースコードのオプティマイザで自動的に最適化したものだという。これでドライストン値を測ると，通常の1.5倍から2倍の値が出る（わがXVIでは3639)。コンパイラにはまだまだ頑張る余地が残っているということか。（KO）

▶先日，(浦)君に連れられて(E.O.)嬢とカイロプラクティックに行ってきた。とりあえず姿勢矯正を軽くやってもらうつもりだったのだが，「あ〜，ずいぶんと曲がってますね。」「ちょっと痛いけどがまんしてね。」と，もみくちゃにされること30分。無事終わり，妙に姿勢正しくなった自分を見てちょっと嬉しいものがあった。また行こうっと。（J）

▶14MHzになった（U）氏のAMIGAに対抗すべく，家のAMIGAに68030ボードを差した。AMIGAに関しては負けてられない。すでにいろんな人に自慢していやがられているが，ここでも自慢しよう。22MHzの68030だ。68882も付いている。動機はもうひとつ，3Dでアニメーションを描こうということだ。もちろん，フライトシミュレータも速くなったけどね。（A）

▶「骨，鳴らしに行こ〜ぜ！」と(浦)に誘われ続けて半年，とうとうカイロプラクティックなるものに行くハメに。ま，いいか，某誌で「痛い怖いの先入観は間違い」って記事があったし……。が，そりゃ大ウソ！　涙ちょちょぎれるほどめちゃ痛いぜよ。その日1日身体はだるく，次の日も背骨が熱くてこれが健康の証なのねと実感したのでした。（E.O.）

▶机の上のEXPERT-HDの総稼動時間が10000時間を超えた。Indy500のベストラップは37.4秒台になった。食事に占める麺類の率が8割近くなった。さてと，これから2カ月間特集は(A)君に任せて，私の割り当てをぐっと減らして，……うーむ，かつてない解放感。当分のあいだ地下工作に専念するとしますか。（U）

▶私が編集部に入ったのが1983年6月号からだから今号で100冊目にあたる。よく続いたものだ。概して真面目にやっているつもりだけど，Oh！Xの正しい読み方(私と(U)で書いたものだ)などを見ると，Oh！Xって結構意地悪だなあと思う。ちなみに，善ちゃんから「いや〜ん」とかも載せてくださいといわれた。だったら善司語録でも作るか。（T）

microOdyssey

机の上が雑然として落ち着いて仕事ができない。いかにして片づけるか，そして片づいた机はいかなるものとなるのか。そのイメージのシミュレーションに日々の時間が費されてしまう。というわけで私が机の上で仕事をすることはあまりない。

さて，雑貨が売れているそうだ。週刊SPAの特集を見ると，売り上げが前年の20～30%増しの百貨店，ファッションビルがザラにあるとのこと。具体的な数字を出されると，どういった商品が雑貨になるのかが気になる。

由緒正しい雑貨といえば，桶，洗剤，亀の子たわしなどだが，いつしか雑貨は実用品としての物から脱皮し，デザインやセンスに重きが置かれるモノへと変化してきた。雑貨のデパートともいえる西武のLOFTには，旧式の「ブタ蚊とり」や，スターモビールならぬ「クジラモビール」など怪しいモノがいっぱいだ。もちろん，そうした特定の品物が売れているわけではない。ここで問題にしている雑貨とは，身のまわりのすべてのものは雑貨であるという大胆不敵なコンセプトに基づく商品体系である。

たとえば，2年前に百貨店の新しい姿を主張して生まれ変わったのが銀座の松屋。何が変わったかというと，1階に雑貨フロアを作り，バッグ，靴，化粧品が伸びているという。ちょっと疑問を感じないだろうか。百貨店の1階というと，もともとそういう品物が売られていたはずだ。だいいち，鞄や靴や化粧品は雑貨なのだろうか。ためしに，化粧品店へ行って「これって雑貨ですか」と聞くと結構いやがられる。

結論をいうと，どんな商品でも単体では雑貨とは呼ばれない。ソニープラザで売っているアプリコットティーも，食料品店へ行けば紅茶の一種でしかない。それが，アリスのティーカップや，罫線幅の広い日記帳，柄のまっすぐなハブラシなどと同じフロアに並んだときに，突如雑貨に変身するのである。

もちろん，一見無関係な商品を結びつけるには，なんらかのテーマ性が必要になる。それは一種の編集感覚である（ここでは雑貨感覚といってもよいだろう）。モノには基本的な属性がある。食器であるとか，時計だとか，傘であるとかだ。従来型ショップではこうした属性を重視する縦割りの座標軸を持っていた。そこに，モノの色だとか形だとか使われ方とかいった質的な属性に着目し，新たな座標軸を構築するのが，雑貨感覚だ。

あるテーマ性を持った雑貨，編集された雑貨は，お客に対し，モノに付加された情報を与える。同じコーヒーカップでも，それがスーパーの食器売り場に置かれた場合と，東急ハンズに置かれた場合とでは商品としての意味が違ってくるのだ。お客は，その店の持つ雑貨感覚をフィルターとして，そこにあるモノに他の店では気づかなかった魅力を見出すわけだ。それは自分自身の指向性を認識することかもしれない。

今回は趣味の話になってしまった。机の上が片づかないので，私はSX-WINDOWのデスクトップで仕事をする。ワープロ，表計算などの大物ソフトはほとんどない。でも，エディタを中心に，便利なアクセサリ類を集め，雑貨感覚で自分なりのデスクトップを作っていく。それはとても楽しいことだ。（T）

1991年10月号 9月18日(水)発売

特集 鉄壁の布陣! ABCD包囲網

A:究極言語アセンブラ　B:外部関数によるBASICの強化
C:C言語との怪しい関係　D:デバッガからの攻略法

新連載 Creative Computer Music入門

Roland GS規格とはなにか?

全機種共通システム
Small-C入門編とデバッグ

バックナンバー常備店

東京	神保町	三省堂神田本店5F 03(3233)3312
	〃	書泉ブックマートB1 03(3294)0011
	〃	書泉グランデ5F 03(3295)0011
	秋葉原	T-ZONE 7Fブックゾーン 03(3257)2660
	八重洲	八重洲ブックセンター3F 03(3281)1811
	新宿	紀伊国屋書店本店 03(3354)0131
	高田馬場	未来堂書店 03(3200)9185
	渋谷	大盛堂書店 03(3463)0511
	池袋	リブロ池袋店 03(3981)0111
	〃	西武百貨店9F コンピュータ・フォーラム 03(3981)0111
神奈川	横浜	有隣堂横浜駅西口店 045(311)6265
	〃	有隣堂ルミネ店 045(453)0811
	藤沢	有隣堂藤沢店 0466(26)1411
神奈川	厚木	有隣堂厚木店 0462(23)4111
	平塚	文教堂四の宮店 0463(54)2880
千葉	柏	新星堂カルチェ5 0471(64)8551
	船橋	リブロ船橋店 0474(25)0111
	〃	芳林堂書店津田沼店 0474(78)3737
	千葉	多田屋千葉セントラルプラザ店 0472(24)1333
埼玉	川越	黒田書店 0492(25)3138
	川口	岩渕書店 0482(52)2190
茨城	水戸	川又書店駅前店 0292(31)0102
大阪	北区	旭屋書店本店 06(313)1191
	都島区	駸々堂京橋店 06(353)2413
京都	中京区	オーム社書店 075(221)0280
愛知	名古屋	三省堂名古屋店 052(562)0077
	〃	パソコンΣ上前津店 052(251)8334
	刈谷	三洋堂書店刈谷店 0566(24)1134
長野	飯田	平安堂飯田店 0265(24)4545
北海道	室蘭	室蘭工業大学生協 0143(44)6060

定期購読のお知らせ

Oh!Xの定期購読をご希望の方は綴じ込みの振替用紙の「申込書」欄にある「新規」「継続」のいずれかに○をつけ，必要事項を明記のうえ，郵便局で購読料をお振り込みください。その際渡される半券は領収書になっていますので，大切に保管してください。なお，すでに定期購読をご利用の方には期限終了の少し前にご通知いたします。継続希望の方は，上記と同じ要領でお申し込みください。

海外送付ご希望の方へ

本誌の海外発送代理店，日本IPS(株)にお申し込みください。なお，購読料金は郵送方法，地域によって異なりますので，下記宛必ずお問い合わせください。

日本IPS株式会社
〒101 東京都千代田区飯田橋3-11-6
☎03(3238)0700

9月号
■1991年9月1日発行　定価600円(本体583円)
■発行人　孫　正義
■編集人　橋本五郎
■発売元　ソフトバンク株式会社
■出版事業部　〒108 東京都港区高輪2-19-13 NS高輪ビル
Oh!X編集部　☎03(5488)1309
出版営業部　☎03(5488)1360 FAX 03(5488)1364
広告センター　☎03(3297)0181
■印　刷　凸版印刷株式会社

ソフトバンクの書籍特約書店

〈東日本編〉

下記の書店は、ソフトバンクの書籍特約書店として,, 右にある商品のほか、新刊も豊富に取り揃えております。ご希望の商品がある場合には、下記の書店にてお買い求め下さい。

＊なお、現品が売れて補充中の場合もありますのでご注意下さい。

ソフトバンク出版事業部
〒108 東京都港区高輪2-19-13 NS高輪ビル
☎03（5488）1360

全国特約書店一覧

市町村	書店名	電話番号
〈北海道〉		
札幌市	紀伊國屋書店札幌店	011-231-2131
〃	東京旭屋書店札幌店	011-241-3007
〃	冨貴堂札幌パルコ店	011-214-2303
〃	ダイヤ書房本店	011-712-2541
〃	ダイヤ書房西店	011-665-6223
〃	ダイヤ書房東店	011-783-3520
〃	丸善らがあーる新札幌ＤＵＯ店	011-890-2588
〃	弘栄堂書店地下鉄店	011-221-5510
〃	本の店岩本北野店	011-883-8036
〃	明正堂１２号線	011-898-0010
〃	本の店岩本平岸店	011-842-0678
小樽市	紀伊國屋書店小樽店	0134-33-1381
旭川市	旭川冨貴堂	0166-26-3481
〃	ブックス平和マルカツ店	0166-23-6211
〃	三省堂書店旭川店	0166-22-6411
苫小牧市	旭屋書店苫小牧店	0144-36-5185
室蘭市	丸善ブックプラザ２１	0143-45-2777
函館市	森文化堂	0138-23-3238
〃	魁文舎	0138-23-0760
滝川市	三光書店	0125-22-1765
帯広市	ザ本屋さん柏林台店	0155-33-5020
〃	田村書店本店	0155-24-5424
北見市	福村書店	0157-23-3330
〈東　北〉		
青森市	成田本店	0177-23-2431
〃	岡田書店	0177-23-1381
弘前市	紀伊國屋書店弘前店	0172-36-4511
〃	メディア・イン城東	0172-27-8118
〃	今泉本店	0172-32-2231
八戸市	金入番町店	0178-46-1811
〃	伊吉書院	0178-44-1917
盛岡市	さわや書店	0196-53-4411
〃	第一書店	0196-53-3355
北上市	小田島書店	0197-63-3351
一関市	北上書房	0191-23-4088
秋田市	三浦書店	0188-33-8131
〃	加賀谷書店	0188-32-3025
大館市	又久書店	0186-42-0305
山形市	八文字屋本店	0236-22-2150
〃	こまつ書店寿町店	0236-41-0641
米沢市	大正堂金池	0238-22-8911
酒田市	みずほ八文字屋	0234-22-3144
〃	青山堂中央店	0234-23-5153
鶴岡市	内山ふみや書店	0235-22-1060
仙台市	金港堂	022-225-6521
〃	金港堂ブックセンター	022-223-0979
〃	金港堂泉店	022-373-4373
〃	アイエ書店駅前店	022-264-0718
〃	アイエ書店	022-221-7268
〃	丸善一番町店	022-266-1127
〃	高山書店	022-263-1511
〃	ブックスみやぎ	022-267-4422
〃	ブックスなにわ中野店	022-258-3133
〃	宝文堂本店	022-222-4181
〃	協裕堂２８６	022-243-2861
仙台市	パソコンＤａｃ仙台東口店	022-291-4744
福島市	岩瀬書店コルニエツタヤ店	0245-21-2101
〃	西沢書店	0245-22-0161
〃	博向堂書店本店	0245-21-1161
会津若松市	宝文館	0242-27-5198
郡山市	東北書店	0249-32-0379
いわき市	ヤマニ書房本店	0246-23-3481
〃	鹿島ブックセンター	0246-28-2222
原町市	文芸堂	0244-22-1720
＜関　東＞		
水戸市	川又書店駅前店	0292-31-0102
〃	ツルヤブックセンター	0292-25-2711
日立市	田所書店	0294-22-5537
〃	一誠堂書店	0294-33-0303
東海村	大野書店	0292-82-2098
勝田市	武石書店	0292-73-1212
鹿島郡	文教堂鹿島店	0299-96-6125
〃	マルエスかみす店	0299-92-1233
〃	なみき書店	0299-96-1855
土浦市	共栄堂	0298-21-6134
〃	白石書店	0298-22-0785
つくば市	ブックランドカスミ学園店	0298-52-3430
〃	丸善筑波大学会館店	0298-51-6000
〃	友朋堂吾妻本店	0298-52-3665
〃	友朋堂梅園店	0298-51-1161
古河市	かもじや本店	0280-22-5700
宇都宮市	落合書店オリオン店	0286-34-3777
〃	落合書店東武ブックセンター	0286-34-8271
〃	新星堂宇都宮店	0286-33-2337
〃	須原屋宇都宮店	0286-27-8351
小山市	進駸堂駅ビル店	0285-25-1522
前橋市	煥乎堂	0272-23-1211
〃	リブロ前橋店	0272-34-1011
〃	戸田書店前橋店	0272-61-5063
〃	文真堂下小出店	0272-33-3381
高崎市	学陽書房	0273-23-4055
〃	サカ井書店	0273-62-1500
〃	新星堂高崎店	0273-27-3961
〃	戸田書店高崎店	0273-63-5110
太田市	ナカムラヤ	0276-22-2001
＜首都圏＞		
浦和市	須原屋本店	048-822-5321
〃	須原屋コルソ店	048-824-5321
大宮市	押田謙文堂	048-641-3141
〃	ブックセンター押田	048-647-3141
大宮市	三省堂ブックポート	048-646-2600
〃	新栄堂書店大宮店	0486-44-2345
蕨市	須原屋蕨店	0484-44-1211
川口市	岩渕書店川口店	0482-52-2190
〃	文教堂東川口店	0482-94-5860
川越市	黒田書店川越店	0492-25-3138
〃	紀伊國屋書店川越丸広店	0492-25-1380
所沢市	芳林堂書店所沢店	0429-25-5355
〃	いけだ書店所沢店	0429-28-3271
〃	パルコ新所沢店	0429-98-8175
飯能市	田中一誠堂	04297-4-1111
入間市	文教堂入間店	0429-63-9481
〃	城西堂鶴ケ島店	0492-86-1161
上福岡市	黒田書店上福岡店	0492-66-0120
狭山市	文教堂狭山店	0429-59-9500
朝霞市	文教堂朝霞店	0484-76-0107
志木市	新星堂志木店	0484-74-0182
新座市	文教堂新座店	0484-79-7070
東松山市	ブックスふじみ	0493-23-0745
秩父市	時習堂	0494-22-1510
比企郡	錦電サービス	0492-96-2962
春日部市	文教堂春日部店	0487-52-7666
〃	ブックガーデン戸田春日部店	048-735-6833
〃	ﾌﾞｯｸｶﾞｰﾃﾞﾝ戸田せんげん台店	0487-35-6833
越谷市	文教堂東越谷店	0489-66-6303
〃	文教堂越谷店	0489-78-5230
千葉市	多田屋セントラルプラザ店	0472-24-1333
〃	キディランド千葉店	0472-25-2011
〃	文教堂小倉台店	0472-32-7330
〃	改造社	0472-25-1333
市原市	文教堂市原店	0436-43-0979
君津市	杉浦書店	0439-52-0159
木更津市	ＢＯＯＫＳ松田屋	0438-23-4210
習志野市	厳翠堂	0474-72-5011
〃	ＢＯＯＫＳ津田沼	0474-77-2610
〃	昭和堂津田沼店	0474-79-2377
八千代市	文教堂八千代台店	0474-83-5622
船橋市	ときわ書房本店	0474-24-0750
〃	リブロ船橋店	0474-25-0111
〃	東京旭屋書店船橋店	0474-24-7331
〃	芳林堂書店津田沼店	0474-78-3737
〃	第二厳翠堂	0474-65-0926
〃	三省堂書店西船橋店	0474-34-3111
〃	紀伊國屋書店船橋店	0474-34-1081
市川市	大杉書店	0473-25-1158
浦安市	原勝書店	0473-51-3920
〃	文教堂浦安店	0473-54-5610
〃	アークブックセンター	0473-54-4623
柏市	西口アサノ	0471-44-2111
〃	新星堂柏店	0471-64-8551
〃	浅野書店スカイプラザ店	0471-64-2040
松戸市	新松戸ブックセンター	0473-41-7030
〃	堀江良文堂	0473-65-5121
〃	辰正堂駅ビル店	0473-64-7997
横浜市	有隣堂トーヨー店	045-311-6265
〃	有隣堂東口ルミネ店	045-453-0811
〃	栄松堂相鉄ジョイナス店	045-321-6831
〃	そごうブックセンター	045-465-2111
〃	丸善ブックメイツポルタ店	045-453-6811
〃	有隣堂伊勢佐木店	045-261-1231
〃	有隣堂たまプラーザ店	045-903-2191
〃	文教堂新横浜店	045-474-3535
〃	有隣堂戸塚店	045-881-2661
〃	文華堂戸塚店	045-864-5151
〃	文教堂青葉台南口店	045-983-5150
〃	文教堂港北ニュータウン店	045-941-6540
〃	アーバン文華堂	045-821-5151

特約書店基本図書一覧

分類	書名	価格
ハード	改訂98NOTEスーパーブック	●2,300円
	ダイナブック・パワーガイド	●1,800円
	マッキントッシュまるかじり	●2,400円
	最新LASER SHOT活用ブック	●2,400円
	X68000マシン語プログラミング	●2,800円
	増設メモリ活用ブック1	●1,900円
OS	新MS-DOS入門 ビギナー編	●1,900円
	新MS-DOS入門 シニア編	●2,300円
	新MS-DOS入門 応用編	●2,300円
	新MS-DOSいたれりつくせり本	●1,900円
	MS-DOSのこわい話	●2,980円
	MS-DOSって不親切?	●1,900円
	Hyper MS-DOS	●2,980円
	MS-DOSハンドブックDynaBook版	●1,500円
	MS-DOS・OS/2コマンドブック	●3,800円
	みるみるわかるMS-DOS ◆FORMAT編◆環境設定編 ◆ディレクトリ編◆バッチファイル編	●各1,400円
	UNIX日記	●1,600円
	Windowsブック	●2,300円
C言語	C言語の応用50例	●2,370円
	上級・C言語の応用50例	●2,480円
	yaccによるCコンパイラプログラミング	●3,300円
C言語	Play the C(上・下)	●各1,550円
	Turbo C Ver.2.0プログラミング	●2,900円
	Quick C Ver.2.0プログラミング	●2,900円
	MS-C Ver.6.0関数リファレンス	●2,000円
	C++	●3,600円
	Cによるプログラミング・スタイルブック	●2,300円
	新C言語入門 ビギナー編	●1,900円
	秘伝C言語問答 ポインタ編	●2,600円
ワープロ	一太郎Ver.3ガイド	●2,580円
	入門一太郎dash J-3100SS	●2,300円
	入門一太郎dash PC-9800版	●2,300円
	4週間で覚える一太郎Ver.4+dash	●2,800円
	PI EXEガイド	●2,600円
	Z'S word J G Ver2.0ガイド	●2,900円
	VZエディターハンドブック	●3,200円
アプリケーション	Lotus1-2-3 R2.2J入門	●2,000円
	Q&A Lotus1-2-3 R2.2J	●2,200円
	最新Lotus1-2-3 R2.2Jガイド ◆マクロ/プログラミング編	●2,900円
	◆マクロライブラリ編	●2,900円
	◆パワーユーザー編	●2,600円
	Lotus1-2-3 Quick Reference	●2,500円
	最新Multiplanガイド Ver.4.1	●2,900円
アプリケーション	Multiplan Ver.4.1演習ノート	●2,000円
	MS-Chart Ver.3.1ガイド	●2,990円
	入門Ninja 3 PROプログラミング	●2,900円
	桐Ver.3 Quick Reference	●3,400円
	桐Ver.3事典 Vol.1 会話処理編	●2,800円
	桐Ver.3ガイド ビギナー編	●2,400円
	アシストカルクガイド	●2,600円
	花子Ver.2ガイド	●2,900円
	DynaCADガイド チュートリアル編	●3,500円
	クイックマニュアルWorks	●2,800円
	Worksガイド ビギナー編	●2,300円
	エコロジーII	●2,200円
	ノートン・ユーティリティーズ	●2,200円
	入門Net Ware	●2,300円
情報処理試験	情報処理入門I	●1,240円
	情報処理入門II	●1,240円
	情報処理入門III	●1,400円
	第2種 ハードウェア徹底マスター	●2,580円
	第2種 ソフトウェア徹底マスター	●2,500円
	第2種 FORTRAN徹底マスター	●2,890円
	第2種 COBOL徹底マスター	●2,990円
ゲーム	RPG幻想事典	●1,550円
	RPG幻想事典・日本編 ジャパネスク	●1,860円

地域	書店名	電話番号
横浜市	住吉書房日吉店	045-561-8855
〃	ブックポート２０３	045-505-0203
〃	トヨムラ横浜店	045-641-7741
川崎市	有隣堂アゼリア店	044-245-1231
〃	有隣堂川崎ＢＥ店	044-200-6831
〃	文学堂本店	044-244-1251
〃	文教堂溝の口店	044-811-8258
〃	文教堂小杉店	044-711-0018
〃	文教堂新城店	044-755-6901
〃	文教堂麻生店	044-989-3160
〃	文教堂宮前平店	044-855-2583
〃	住吉書房小杉店	044-711-2121
〃	向文堂	044-922-7600
〃	住吉書房登戸店	044-932-1000
〃	大塚書店百合ケ丘店	044-954-5281
〃	流水書房ＫＳＰ店	044-819-2313
鎌倉市	島森書店大船店	0467-46-3841
〃	鎌倉書店	0467-46-2619
〃	文教堂鎌倉店	0467-31-8961
横須賀市	平坂書房ＷＡＬＫ店	0468-25-5537
藤沢市	有隣堂藤沢店	0466-26-1411
〃	リブロ藤沢店	0466-27-0111
〃	文教堂六会店	0466-82-9610
〃	文教堂辻堂店	0466-33-6288
茅ケ崎市	川上書店ルミネ店	0467-87-3827
平塚市	サクラ書店駅ビル店	0463-23-2751
〃	文教堂四之宮店	0463-54-2880
小田原市	八小堂書店	0465-22-7111
〃	伊勢治書店	0465-22-1366
〃	文教堂小田原店	0465-36-3677
厚木市	有隣堂厚木店	0462-23-4111
大和市	文教堂中央林間店	0462-75-4165
相模原市	文教堂相模大野店	0427-49-0650
〃	文教堂相模台店	0427-48-8144
〃	文教堂橋本店	0427-74-5581
〃	アクトブックス相模原店	0427-56-5628
〃	文教堂星ケ丘店	0427-58-6121
秦野市	伊勢原書店	0463-83-0841
海老名市	三省堂書店海老名店	0462-34-7161
津久井郡	文教堂城山店	0427-82-9278
〃	伊勢原書店城山店	0427-82-8824
中郡	文教堂平塚店	0463-31-3911
<東　京>		
千代田区	三省堂書店本店	03-3233-3312
〃	書泉グランデ	03-3295-0011
〃	東京堂書店	03-3291-5181
〃	飯田橋書店	03-3263-5401
〃	山王麹町店	03-3264-5581
〃	東京旭屋書店水道橋店	03-3294-3781
〃	丸善お茶の水店	03-3295-5581
〃	巌翠堂	03-3291-1361
〃	いずみ神田南口店	03-3254-8521
〃	明正堂秋葉原店	03-3257-0758
〃	Ｂｉｔ ＩＮＮ 東京	03-3255-4575
〃	Ｔ－ＺＯＮＥ	03-3257-2660
〃	ラオックス THE COMPUTER館	03-5256-3111
千代田区	秋葉原ブックセンター	03-3255-0551
〃	二六堂岩本町店	03-3254-1108
中央区	八重洲ブックセンター	03-3281-1811
〃	丸善日本橋本店	03-3272-7211
〃	東京旭屋書店銀座店	03-3573-4936
港区	書原新橋店	03-3591-8738
〃	雄峰堂書店ＮＳ店	03-3503-6586
〃	虎ノ門書房本店	03-3502-3461
〃	虎ノ門書房田町店	03-3454-2571
〃	誠志堂東日ビル店	03-3405-2889
品川区	芳林堂書店大井町店	03-3474-4946
〃	明屋書店五反田店	03-3492-3881
〃	ソーブン堂	03-3783-5801
渋谷区	紀伊國屋書店渋谷店	03-3463-3241
〃	東京旭屋書店渋谷店	03-3476-3971
〃	三省堂書店渋谷店	03-3407-4545
〃	大盛堂書店	03-3463-0511
〃	紀伊國屋書店笹塚店	03-3485-0131
新宿区	紀伊國屋書店本店	03-3354-0131
〃	三省堂書店新宿西口店	03-3343-4871
〃	福家書店センタービル店	03-3345-1246
〃	福家書店野村ビル店	03-3342-0298
〃	新星堂ＮＳビル店	03-3344-2055
〃	西武新宿ブックセンター	03-3208-0380
〃	芳林堂書店高田馬場店	03-3208-0241
〃	未来堂	03-3200-9185
〃	ラムラブックセンター芳進堂	03-3235-5111
〃	あゆみブックス早稲田店	03-3203-7123
〃	博文堂書店本店	03-3359-5301
〃	ＬＡＯＸ新宿店	03-3350-1241
豊島区	芳林堂書店本店	03-3984-1101
〃	東京旭屋書店池袋店	03-3986-0311
〃	リブロ池袋店	03-3981-0111
〃	三省堂書店池袋店	03-3987-0511
〃	新栄堂本店	03-3984-2345
〃	新栄堂アルパ店	03-3988-0181
練馬区	ＬＩＣ英林堂	03-3921-6521
〃	リブロ光ケ丘	03-3976-8111
板橋区	博文堂書店大山店	03-3963-4001
〃	高島平南天堂	03-3935-6227
台東区	明正堂中通り店	03-3831-0191
台東区	リブロ浅草店	03-3836-7800
墨田区	ブックストア談	03-3635-1841
〃	リブロ錦糸町店	03-3846-0111
江東区	新栄堂書店亀戸駅ビル店	03-3638-2345
江戸川区	文教堂西葛西店	03-3689-3621
葛飾区	文教堂青戸店	03-3838-5931
〃	第一書林	03-3653-2421
大田区	アクトブックスサンカマタ店	03-3735-1551
〃	竜文堂大森駅ビル店	03-3775-3851
〃	栄松堂書店蒲田店	03-3731-2244
目黒区	恭文堂書店	03-3712-4049
〃	三省堂書店自由ケ丘店	03-3718-2108
中野区	明屋書店東京本店	03-3387-8451
杉並区	ブックセンター荻窪	03-3393-5571
〃	ブックセラーズ西荻	03-3395-6566
杉並区	書原杉並店	03-3313-4778
武蔵野市	紀伊國屋書店吉祥寺東急店	0422-21-5543
〃	弘栄堂吉祥寺店	0422-22-1031
〃	パルコブックセンター吉祥寺	0422-21-8122
調布市	真光書店	0424-87-2222
〃	パルコブックセンター調布店	0424-89-5351
府中市	啓文堂	0423-66-3151
三鷹市	三省堂書店三鷹店	0422-48-4510
〃	東西書房	0422-46-0275
小金井市	文教堂小金井店	0423-86-0161
〃	文教堂新小金井店	0423-82-8885
国分寺市	三成堂国分寺店	0423-25-3211
〃	三石堂書店	0423-21-0969
国立市	東西書店	0425-75-5061
〃	増田書店	0425-72-0262
田無市	田無書店	0424-66-0361
小平市	文教堂小平店	0423-43-9229
東村山市	文教堂東村山店	0423-96-1115
立川市	オリオン書房ウイル店	0425-27-2311
青梅市	ブックス・タマ	0428-24-2450
八王子市	くまざわ書店八王子本店	0426-25-1201
〃	朋隣堂	0426-42-2763
町田市	有隣堂町田店	0427-23-3018
〃	久美堂本店	0427-25-1330
〃	久美堂小田急店	0427-25-1111
〃	文教堂鶴川店	0427-35-4117
〃	文教堂小川店	0427-96-1781
〃	文教堂成瀬店	0427-28-0936
〃	文教堂南成瀬店	0427-24-6430
多摩市	くまざわ書店桜ケ丘店	0423-37-2531
〃	紀伊國屋書店そごうﾌﾞｯｸｾﾝﾀｰ	0423-39-2474
〃	リブロ多摩店	0423-38-6221
〃	啓文堂センター	0243-72-2288
福生市	文教堂福生店	0425-53-7708
保谷市	正育堂書店	0424-22-0434
東久留米市	文教堂東久留米店	0424-71-8188

赤えんぴつ (JRA版)

X68000で競馬の予想が出来る。

この『赤えんぴつ』はJRA(日本中央競馬会)の競馬場で催されるレース専用の競馬予想プログラムです。

『赤えんぴつ』は当たる馬券を予想するのでは無く、新聞に掲載されている過去のデータを、各馬毎に入力してゴールした時のタイムを予想して、当たり馬券を割り出すので、今までに無い当たり馬券の予想が期待出来ます。

★過去のデータだけを入力するのでは無く、最新の馬の調子や、あなたのお考え等を10～100%の数値に置き換えて予想に反映させる事が出来ます。

★入力したデータをディスクにセーブする事が出来ますのでレースの前日にデータを入力して、当日手直しして予想を立て直す事も出来ます。

★予想の結果をプリンターへ印刷する事が出来ます。

★キーボードの苦手な方でもカーソルコントロールキーと実行キーの5つのキーを使うだけでデータの入力が出来、キーボードを使い慣れた方は殆どのデータがテンキーから直接入力する事も出来ます。

【有料モニターの募集】

『赤えんぴつ』を一度試してみたい方に4,000円で『赤えんぴつ』から一部の機能(データのセーブ・ロード、プリンターへの出力等)を除いたモニター版の『赤えんぴつ』をお送りします。
お試し頂き、本来の『赤えんぴつ』をご希望の方には16,000円で『赤えんぴつ』をお求め頂けます。
尚、この有料モニター・サービスは当社へ直接4,000円を同封してお申し込み下さい。

・このプログラムは、JRA主催の全国10ヶ所の各競馬(札幌、函館、福島、新潟、中山、東京、名古屋、京都、阪神、小倉)以外の公営競馬場では使えません。

'91年版データ付き

赤えんぴつ	X68000用	2HD	20,000円

＊商品の価格には消費税は含まれていません。

▶お求めは全国の有名マイコンショップでどうぞ。
通信販売をご希望の方は当社へ直接、商品名・機種名・メディア名・住所・氏名・電話番号を明記の上、現金書留にてお申し込みください。(送料無料)

BLUE SKY Co.
株式会社 BLUE SKY
〒411 静岡県三島市加茂16-4 ☎0559-72-6710

満開の電子(でんこ)ちゃん

作・え　岡村祭

購読方法：定期購読もしくはソフトベンダー武尊(タケル)でお買い求めいただけます。
★定期購読の場合＝定期購読料６ヶ月分6,000円（送料サービス、消費税込）を、
　現金書留または郵便振替で下記の宛先へお送り下さい。
　　現金書留の場合：〒171　東京都豊島区要町1―19―3　いさみビル4F　満開製作所
　　郵便振替の場合：東京5―362847　満開製作所
　●御注文の際は、郵便番号・住所・氏名・電話番号を忘れずに記入して下さい。
　●新たに購読を開始される方は、「新規」とご明記下さい。
　●製品の性格上返品には応じられませんが、お申し出があれば定期購読を解約し残
　　金をお返しします。
★武尊でお求めの場合＝１部につき1,200円（消費税込）です。
　●定期購読版と内容が一部異なる場合があります。ご了承下さい。
●お問い合わせ先　TEL(03)3554―9282（月～金　午前11時～午後６時）
(なお、定期購読版のバックナンバーについては定期購読者の方のみご注文を承ります)

鳥居秀則（京都府）

どんなに忙しくても、またどんなに疲れていても、水色の封筒が家に届いた時だけ私は別人になります（写真参照）。もちろんその封筒の中身は電脳倶楽部です。読み物・ツール・ビープ音・それにQ&Aコーナーなどに代表されるように、とにかく役に立つデータがいっぱいで、それらがすぐ実行できるようにディスクに入っています。怠慢を絵にしたような私にとってこれ以上のものはありません。これでひと月千円で、どうして儲けがとれるのか、などと余計な心配をしてしまうのは私だけでしょうか？（やっぱり怪しい）

新作 X68000用日コン連SOFT

関西学院大学Ｌ．Ｅ．Ｃ．のデビュー作

Indulging Puzzle Game

Loop Eraser

(ループ イレーサー)

5,980円

画面内の２種類のパーツを回転させて、ループを作り、ループを構成しているパーツを消去。画面内のパーツを全て消去すれば面クリアというパズルゲームです。

全９９面。オリジナル面ためのコンストラクション機能付きです。

第５回オールジャパンオリジナルソフト博覧会人気第１位獲得作品。

関西学院大学電脳研究会のデビュー作

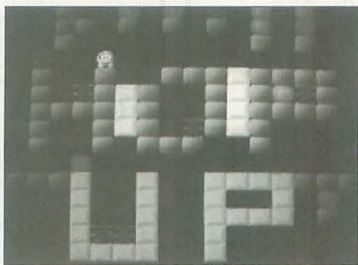
Exciting Puzzle Game

HOP UP

(ホップ アップ)

5,980円

赤いブロックを赤いジャンプ台に。青いブロックを青いジャンプ台にすべて移すと面クリア。全100面。オリジナル面作成のためのコンストラクション機能付きです。

第４回オールジャパンオリジナルソフト博覧会人気第１位獲得作品。

神戸大学情報統計部Ｄ＿ＲＥＴＵＲＮ赤坂賢洋の第２弾

究極のシミュレーションゲーム

PLANETARY CAMPAIGN 68K

(プラネタリー キャンペーン)

4,980円

太陽系を舞台とした壮大なシミュレーションゲーム。

赤坂賢洋の自信作品。

ベストセラーを狙ったアイデアソフト決定版

ずるかまし辞書のスペル完全マスターツール

スペルマスターかきたおし　5,980円

スペルマスタージュニアかきたおし　4,980円

スペルマスターかきたおしには、翻訳ヘルパーずるかまし収録の大学入試用の約５０００語、スペルマスタージュニアかきたおしには、高校入試用の約１５００語の英単語を収録しています。マウスのみを用いて、ゲーム感覚で楽しくスペルのマスターが行えます。スペルマスターかきたおしには、受験勉強の息抜きのためにエッチな英単語ばかり集めたアダルト辞書をおまけにつけています。

好評発売中！

Ｘ６８０００用 日コン連ＳＯＦＴ

アドベンチャーゲームが簡単に作れる電脳作家シリーズ

電脳作家Ver.2.0　5,980円

電脳作家グラフィック&ミュージックライブラリー集　3,980円

電脳作家シナリオ集１　2,980円

究極のシューティングゲーム

ＴＯＷＮＳ版は、全ソースリスト公開！

D-RETURN　5,980円

神戸大学情報統計部の赤坂賢洋がたった一人で作成、大ヒットを飛ばした伝説のゲーム。

ワクチンソフトのベストセラー

サイバーワクチンいてこまし　3,000円

X68000のすべてのウイルスのＳ－ＲＡＭからの除去が可能。

宿題が楽になったと高校生に大人気！

翻訳ヘルパーずるかまし　5,980円

英文翻訳ガイド、英和辞典、和英辞典、英単語暗記トレーニング、辞書ユーティリティ、添付辞書５０００語からなる翻訳の友です。

大阪市立大学マイコン研究会山本賢一のアイデアソフト。

近畿大学附属高校生水無月みるくの作成したＸ１版もあります。

ずるかましジュニア辞書　2,980円

ずるかましの別売辞書。中学生単語約１５００語収録。

ずるかまし辞書とジュニア辞書とのマージプログラム付き。

大阪市立大学マイコン研究会西尾幸人が制作。

通信販売のご案内

日コン連ＳＯＦＴは、すべて通信販売で購入できます。
ご希望の商品名・機種名・媒体名明記の上、郵便振替で、大阪５－４８７３　日コン連企画㈱または、現金書留・定額小為替などで、お送りください。消費税・送料は、サービスさせて頂きます。お釣りは、切手でお返しします。

パソコン&キャンパス雑誌　Ｃ・ａｂｌｅ（ケーブル）案内

日コン連発行の過激雑誌です。Ｊ＆Ｐチェーン等全国８０の書店・パソコンショップで販売されています。入手困難な方は、希望号名と送料込みで創刊号５７０円、２号・３号各７６０円（ディスク付）相当の切手をお送りください。

只今募集中！

日本コンピュータクラブ連盟加盟団体

オリジナル開発ソフト

日コン連本部スタッフ、Ｃ・ａｂｌｅ編集部スタッフ

日本コンピュータウイルス研究学会会員

●お問い合わせ・お申込先

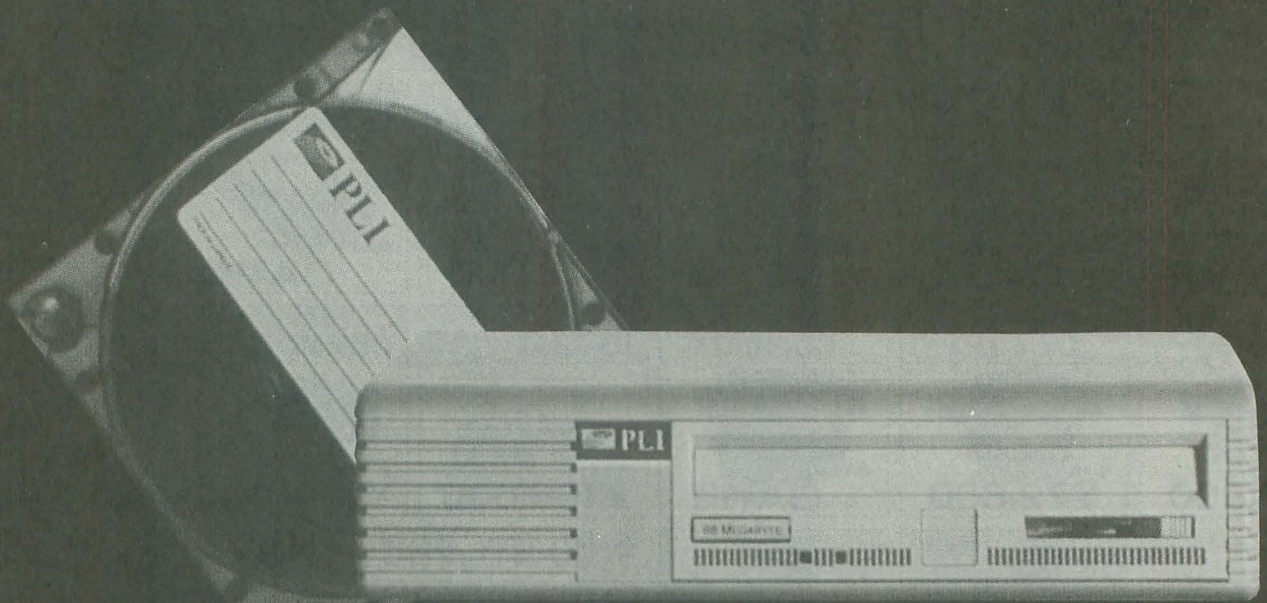
〒５５６　大阪市浪速区難波中２－４－３　村上ビル

日コン連企画㈱または、日本コンピュータクラブ連盟

電話０６－６４４－６９０１（代）

THE Removable HDD

メディア交換可能な新世代ハードディスク

PLI Infinity 40 turbo

Oh!X 特別価格
¥148,000
メディア2枚サービス

CZ-6BS1
セット価格
¥170,000

リムーバブルハードディスクはフロッピーディスクや光磁気ディスクと同じようにメディアを抜き差しできる新世代のハードディスクです。
交換が可能になることによりデータのバックアップなどをメディアごと行なうことも可能ですし、他のInfinity 40 turboユーザーとのデータ交換なども簡単に出来るようになります。
PLI社のInfinity 40 turboはアメリカでマッキントッシュやIBM PC用として既に多大な評価を受けておりその品質は万全です。気になる平均シークタイムも20msecと固定型のハードディスクに引けを取らない高速です。BASIC HOUSEはこの大変便利なデバイスをぜひX68000ユーザーの皆さんにも使用していただきたいと思い、販売を開始いたします。

★平均シークタイム20msec
★メディア1枚当たりの容量42Mバイト
★連続耐久時間30000時間以上
★SCSIインターフェース対応
★X68000用SCSIケーブル、ターミネータ付属
★メディア1枚(40Mバイト)の価格はわずか¥18,000

※ SUPER,XVI以外の機種ではSCSIインターフェースボード(CZ-6BS1)が必要です。

BASIC HOUSE 計測技研

お申し込み・お問い合わせは 0286-22-9811(代)

〒321 栃木県宇都宮市竹林町503-1 FAX 0286-25-3970

AIBIT
アイビット電子株式会社

SHARP

パソコン本体から周辺機器まで品数取り揃え

大特価セール実施中!!

型名	品名	正価	特価
PC-E550	ポケコン64K	32,000	25,600
PC-E500BL	ポケコン	28,800	19,500
PC-1600K	ポケコン	69,800	49,800
PC-1360K	ポケコン	36,800	32,800
PC-1360	ポケコン	29,800	19,800
PC-1262	ポケコン	24,800	19,600
PC-1248DB	ポケコン	11,000	9,800
PC-1280	ポケコン	24,800	19,600
CE-T800	ポケコンRS-232Cコンバーター	12,800	11,800
CE-2H16M(16K)		16,000	11,000
CE-2H32M(32K)		32,000	16,000
CE-2H64M(64K)		45,000	30,000
CE-212M(8K)		18,000	6,000
CE-203M	ポケコンRAM32K	32,000	7,000
CE-202M	ポケコンRAM16K	35,000	6,000
CE-201M	ポケコンRAM 8K	18,000	3,000
CE-1600M	ポケコンRAM32K	32,000	16,000
CE-1600F	ポケコンフロッピードライブ	39,800	34,800
CE-1600P	ポケコンプリンター	69,800	59,800
CE-1650F	ポケコンDISK	9,800	8,800
CE-161	ポケコンRAM16K	50,000	3,800
CE-1601M	ポケコンRAM64K	45,000	30,000
CE-1600E	ポケコンディスクインターフェイス	19,800	17,800
CE-158	ポケコンレベルコンバター	39,800	31,300
CE-159	ポケコンRAM 8K	35,000	4,200
CE-140T	ポケコンRS-232Cコンバター	9,800	8,800
CE-140F	ポケコンフロッピディスク	49,800	44,800
CE-130T	ポケコンRS232Cレベルコンバータ	17,800	15,800
CE-135T	ポケコンRS422レベルコンバータ	9,800	8,800
CE-123P	ポケコンプリンター	19,800	17,800
CE-120P	ポケコンプリンター	24,800	21,800
CE-126P	ポケコンプリンター	17,800	13,800
CE-124	ポケコンカセットインター	4,500	3,600
Z-VISIONplus	Z80シュミレータ デバッカー	59,800	51,000
UX-1	ホームコピーファクス	78,000	69,800
PA-9500	ハイパー電子手帳	48,000	特価
CZ-300F	X13"マイクロフロッピー	79,800	9,000
CZ-31FS	300F増設フロッピー	59,800	7,000
CZ-82F	CZ-802C増設フロッピー	59,800	6,000
CZ-501H	X1増設用ハードディスクユニット	258,000	60,000
CZ-6BS1	SCSIボード	29,800	23,800
CZ-6BP1	数値演算ボード	79,800	63,800
CZ-6BU1	ユニバーサルI/Oボード	39,800	33,800
CZ-6BM1	MIDIボード	29,800	23,800
CZ-6BEIB	1M増設RAMボード	28,000	19,500
CZ-6BEI	1M増設RAMボード	35,000	29,500
CZ-6BE2	2M増設RAMボード	79,800	63,800
CZ-6BE4	4M増設RAMボード	138,000	110,400
CZ-6BN1	スキャナーボード	29,800	25,300
CZ-6BF1	RS-232C増設ボード	49,800	42,300
CZ-6SD1	システムラック	44,800	38,000
CZ-6TU	RRGBシステムチューナー	33,100	26,500
CZ-6BG1	X68000GPIBボード	59,800	50,000
CZ-6BC1	X68000FAXボード	79,800	67,800
CZ-6PV1	ビデオプリンター	198,000	158,000
CZ-6BV1	ビデオボード	21,000	16,800
CZ-822C	X1G MODEL30	118,000	39,800
CZ-820C	X1G MODEL10	69,800	16,800
CZ-128SF	X1CP/M	9,800	8,500
CZ-130SF	X1turbo漢字CP/M	14,800	12,500
CZ-115LF	X1FORTRAN	13,800	11,700

型名	品名	正価	特価
CZ-116LF	X1C	13,800	11,700
CZ-8BGR2	グラフィックボードX1	14,800	3,000
CZ-8BF1	FDインターフェイス	14,800	11,500
CZ-8BK2	X1 漢字ROM	19,800	16,800
CZ-8BM2	232Cマウスボード	19,800	16,800
CZ-8BE2	320K外部メモリー	29,800	25,300
CZ-8BR1	立体映像セット	29,800	25,300
CZ-8BV2	カラーイメージボート	39,800	32,000
CZ-8BO1	FDインターフェイス	14,800	8,000
CZ-8TM1	X1ソフト付モデムユニット	29,800	5,000
CZ-8TM2	X1ソフト付モデムユニット	49,800	39,800
CZ-8EB3	拡張i/o box	33,800	28,000
CZ-8LM1	232cケーブル	7,200	6,000
CZ-8LM2	232cクロスケーブル	7,200	6,000
CZ-8NJ1	ジョイカード	1,700	1,360
CZ-8NT1	トラックボール	13,800	11,500
CZ-8PK10	24ドット136桁漢字プリンター	99,800	大特価
CZ-8PK7	24ドット80桁漢字プリンター	122,000	59,800
CZ-8PC5BK	48ドット熱転写カラー漢字プリンター	96,800	新発売
CZ-8BS1	X1FM音源ボード	23,800	19,500
CZ-8BK4	X1第2水準ROM	———	5,700
CZ-8NJ2	インテリジェントコントローラー	23,800	18,500
CZ-8NS1	カラーイメージスキャナー	188,000	149,000
CZ-612DGY	15インチ0.31TV付	119,800	80,000
AN-S100	アンプ付スピーカー	36,600	29,500
AN-X68	キーボードシリコンカバー	3,500	2,800
AN-X68PRO	キーボードシリコンカバー	3,500	2,800
AN-1508	ディスプレイ15P→8P変換ケーブル	———	1,600
AN-1506	ディスプレイ15P→ 変換ケーブル	———	1,600
HXD040	アイティム40Mハードディスク(ITM)	118,000	89,000
HXD140	40Mハードディスク内蔵用(ITM)	98,000	79,800
CU-14FD	カラーディスプレーアナログ0.31	74,800	49,800
CU-14KD	アナログ0.28 14"CRT	89,800	59,800
CU-14TV	デジタル/アナログ0.31 14"CRT TV付	98,800	64,800
MZ-1D10	12"モノクロディスプレー	41,800	25,000
MZ-1D17	15"CRT mz-5500/6500/2	124,000	59,800
MZ-1E05	MZ-2000 FDインターフェイス	24,500	18,000
MZ-1E08	プリンターI/F 2000/2200/80B	9,000	8,000
MZ-1E11	MZ-6500用 SFD I/F	38,000	25,000
MZ-1E04	MZ-2000 プリンターI/F	10,000	6,000
MZ-1E21	MZ-5500 GP I/F	36,000	12,000
MZ-1E18	MZ2000QD用インターフェイス	9,800	3,000
MZ-1E33	MZ6500パラレルI/F	34,800	28,000
MZ-1E45	MZ6500 232C I/F	50,000	15,000
MZ-1E32	MZ2500 パラレル I/F	30,000	27,000
MZ-1E44	MZ-6500 S-RN I/F	50,000	15,000
MZ-1E22	MZ-5500 GPIB I/F	72,800	25,000
MZ-1E29	RS-232Cインターフェイス300BT	17,800	9,800
MZ-1E01	MZ-3500 232Cボード	28,000	13,000
MZ-1E14	MZ1500 QD用インターフェイス	9,800	3,000
MZ-1M01	MZ-2000/2200 16ビットボード	78,000	8,000
MZ-1M09	MZ-6500 8082-2演算プロセッサ	82,000	30,000
MZ-1M03	MZ-5500 数値演算	69,000	38,500
MZ-1M12	MZ-2861 8087 演算プロセッサ	90,000	45,000
MZ-80P4B	136桁ドットプリンター	———	48,000
MZ-1P06	ドットプリンター	234,000	45,000
MZ-1P27	水平漢字プリンター	268,000	大特価
MZ-1P28	ドットプリンター漢字80桁	148,000	118,400
MZ-1P10A	24ドットプリンター漢字80桁	245,000	79,000
MZ-1P22	熱転写漢字プリンター	59,800	25,000
MZ-1P29	漢字プリンター136桁	168,000	134,400

型名	品名	正価	特価
MZ-1P30	136桁プリンター	228,000	120,000
MZ-1R01	MZ-2000/2200Gボード	39,800	10,000
MZ-1R10	MZ-5500 漢字ROM付	30,000	9,800
MZ-1R09	MZ-5500 V.RAM	35,000	15,000
MZ-1R06	MZ-5500 増設RAM	45,000	8,000
MZ-1R12	MZ-80B/2000/1500/700 RAM	35,000	8,000
MZ-1R11	MZ-5500 256KRAM	80,000	35,000
MZ-1R36	MZ-28611M増設RAM	45,000	15,000
MZ-1R35	MZ-28611M増設RAM	55,000	19,000
MZ-1R14	MZ-5500 辞書ROM	40,000	22,000
MZ-1R16	MZ-5500 128KRAM	30,000	8,000
MZ-1R21	漢字ROM	38,000	13,000
MZ-1R24	MZ-1500 辞書ROM	22,000	6,000
MZ-1R32	MZ-6500RAM	80,000	40,000
MZ-1R31	漢字ROM	28,000	20,000
MZ-1R28A	MZ-2500 辞書ROM	13,000	10,000
MZ-1S13	MZ-1D17チルトスタンド	12,000	5,000
MZ-1T02	MZ-2200 データーレコーダー	19,800	8,500
MZ-1T03	MZ-5500 データーレコーダー	12,000	8,500
MZ-1V01	パソコン FAX	278,000	85,000
MZ-1X22	モデムユニット	21,800	13,000
MZ-2Z016	MZ-5500 附属	———	5,000
MZ-2Z023	MZ-5500 GWBASIC	50,000	30,000
MZ-2Z031	MZ-6500 日本語ワープロ	49,800	15,000
MZ-2Z029	MZ-6500 TODAY	68,000	20,000
MZ-2Z064	MZ-6500 書院RAM付	69,800	28,000
MZ-2Z065	MZ-6500 書院RAMなし	49,800	15,000
MZ-2Z012	MZ-5500 附属	———	5,000
MZ-2Z013	MZ-5500 MSDOS	25,000	20,000
MZ-4Z001	MZ-5500 IBM変換	30,000	8,000
MZ-5521	本体	388,000	55,000
MZ-5511	本体	288,000	35,000
MZ-5Z013	MZ-1500 QD通信ソフト	———	3,500
MZ-6F03	ブランク QD DISK	450	400
MZ-6P18	MZ-1P18,28カットシートフィーダー	60,000	35,000
MZ-6P29	MZ-1P29 カットシートフィーダー	50,000	37,500
MZ-6P27	MZ-1P27 カットシートフィーダー	58,000	39,800
MZ-6P06	MZ-1P06 トラクターフィード	15,000	7,500
MZ-6P20	MZ-1P22/17ロールホルダー	3,100	2,700
MZ-6Z22	MZ-6500(50)CP/M86BASIC-3	10,000	6,000
MZ-6Z25	M-50 ストリーマユーティリティZプロセッサ	39,800	15,000
MZ-80T20A	MZ-80 マシンランゲージ	6,000	5,000
MZ-80TUB	MZ-80 バックアップ	20,000	8,000
MZ-80TU	MZ-80 システムプログラム	20,000	8,000
MZ-80T40A	MZ-80 PASCAL	10,000	5,000
MZ-80T70A	MZ-80 FDOS	20,000	7,000
MZ-8BGK	MZ-80 BGRAM2	39,000	10,000
MZ-8B104	MZ200/2200 GP IBインターフェイス	45,000	18,000
MZ-8BC01	MZ200/2200 GP IBケーブル	18,000	8,000
MZ-1D27デジタル/アナログ0.31 14"CRT TV付 MZ25/28シリーズ用		99,800	69,800
MZ-8376A	AXノート286N	398,000	298,000
IO-735X	カラープリンター	248,000	180,000
BF-68PRO	フィルター	19,800	16,800
X68000キーボード延長ケーブル(1.5m)		2,500	2,000
ディスプレーケーブルアナログ15P(3m)		5,000	4,000
ディスプレーケーブルアナログ15P(1.5m)		4,300	3,500
▶XC-100P	イメージプレゼンテーター	398,000	298,000
▶CU-SX1	パソコン用プロジェクター	980,000	800,000
▶XC-10SC1	24K→15Kスキャンコンバーター	300,000	240,000
▶LEDかんばん	15cm×15cm×8文字3色	950,000	特価

▶印は御注文いただいてから発送まで少々時間がかかります。

ポケコン関係周辺機器サプライ製品及シャープ関係のソフトウエア全種取扱います。

FM TOWNS/FM NOTE/東芝ダイナブック、周辺機器も取扱っております。

AIBIT
アイビット電子株式会社

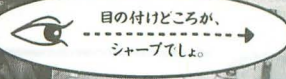
X68000 XVIフェアー!!

大特価セール実施中!

XVIからEXPERTまで在庫あり。即納!

'91年9月15日迄

X68000下取りセール実施中!

目の付けどころが、シャープでしょ。

X68000

X68000セットでお買上げの方にソフト2本(当店指定)プレゼント(9/15まで)

XVI
CZ-634CTN
+CZ-606DTN
アイビット特価

XVI HD
CZ-644CTN
+CZ-606DTN
アイビット特価

EXPERT
CZ-602CGY
+CZ-603DGY ¥248,000
+CZ-612DGY ¥268,000

EXPERT+HD
CZ-602CGY+HXD140
+CZ-603DGY ¥315,000
+CZ-612DGY ¥335,000
+CZ-613DGY ¥355,000

XVI
CZ-634CTN
+CZ-614DTN
アイビット特価

XVI HD
CZ-644CTN
+CZ-607DTN
アイビット特価

EXPERTII
CZ-603C
+CZ-606D ¥270,000
+CZ-607D ¥285,000
+CZ-614D ¥310,000

Super HD
CZ-623CTN
+CZ-606DTN
+CZ-607DTN
+CZ-614DTN

XVI
CZ-634CTN
+CZ-607DTN
アイビット特価

XVI HD
CZ-644CTN
+CZ-613DTN
アイビット特価

Super
CZ-604CTN
+CZ-606DTN
+CZ-607DTN
+CZ-614DTN

PROII
CZ-653CBK
+CZ-606DBK
大特価

X68000周辺機器

ドットマトリクス漢字プリンタ(136桁)
CZ-8PK10
標準価格 ¥97,800
特価

カラーイメージジェットプリンタ
IO-735X-B
標準価格 ¥248,000
特価 ¥198,500

電子手帳データーを自由にカッティング
MC-300
定価 ¥580,000
資料請求して下さい。

X68000 3.5インチフロッピーディスクユニット
X6835-2F
標準価格 ¥80,000
特価

24ピン漢字プリンタ(80桁)
CZ-8PK7
標準価格 ¥122,000
特価 特価

カラーイメージスキャナ
232Cケーブル,スキャナツールソフト付
JX-220X 標準価格¥168,000
特価 ¥134,500

光磁気ディスクユニット
CZ-6M01
標準価格 ¥450,000
特価 ¥360,000

HXD 040 23ms X68000
標準価格¥118,000 特価¥79,800
HXD 042 X68000 増設用
標準価格¥128,000 特価¥102,500
HXD 140 X68000 内蔵用
標準価格¥98,000 特価¥75,000
HXD 140は602C,603C,652C,653Cの内蔵用

X68000ソフト

		正価	特価
ICONEDITER	計測技研	¥ 4,800	¥ 4,100
C言語ライブラリー(X68000)	計測技研	¥ 6,800	¥ 5,800
BASIC拡張関数パッケージ	計測技研	¥ 9,800	¥ 8,500
DISK CACHER	計測技研	¥ 6,800	¥ 5,800
たーみのる2	SPS	¥17,800	¥14,250
X1エミュレータ	ACCESS	¥ 9,800	¥ 8,330
EM 68K(エミュレータ)	ニューウェーブ	¥30,000	¥25,500
CP/M 68K	ニューウェーブ	¥110,000	¥93,500
サイクロン Ver1.2	アンス	¥58,000	¥49,300
SUPER DIVICE MONITOR "T"	ブルースカイ	¥15,000	¥12,000
CANVAS	CZ-249GS	¥29,800	¥23,850
XBAS to C CHECKER	CZ-260LS	¥ 9,800	¥ 7,850
Multiword	CZ-225BS	¥32,000	¥25,600
Human68K Ver2.0	CZ-244SS	¥ 9,800	¥ 7,850
C compiler Ver2.0	CZ-245LS	¥44,800	¥35,850
SOUND PRO68K	CZ-214BS	¥15,800	¥12,650
Sampling PRO68K	CZ-215MS	¥17,800	¥14,250
MUSIC PRO68K	CZ-213MS	¥18,800	¥15,000
MUSIC Studio PRO68K Ver2.0	CZ-261MS	¥28,800	¥23,000
MUSIC PRO68K MIDI	CZ-249MS	¥28,800	¥23,000
CARD PRO68K Ver2.0	CZ-253BS	¥29,800	¥23,850
CARD活用フォーム集1	CZ-242BS	¥ 9,800	¥ 7,850
SX-WINDOW Ver1.1	CZ-278SS	¥ 9,800	¥ 7,850
OS-9	CZ-219SS	¥29,800	¥23,850
Easypaint SX-68K	CZ-263GW	¥12,800	¥10,250
Teleportion PRO68K	CZ-258BS	¥22,800	¥18,250
Cプロフェッショナルパッケージ	マイクロウェア	¥58,000	¥49,300

※富士通、NEC、シャープ周辺機器(拡張機器全機種、プリンター他)も常時取り扱っております。〈全商品新品完全保証付〉 シャープ、カシオポケコン全機種取扱い。カタログ、価格表ご請求には、72円を添えてお願い致します。

通信販売のお問い合せ、御注文は

0426-45-3001(本店)

FAX.0426-44-6002

●営業時間/10:00~19:00●電話受付/20:00迄可●定休日/水曜日

SHARP SUPER XEX SHOP

アイビット電子株式会社 〒192 東京都八王子市北野町560-5

16号バイパス
1Fショップ
2Fパソコン教室
ココ(ご来店は駅前徒歩0分)
至京王八王子
京王北野駅
至新宿
至横浜

上記の広告商品はすべて店頭販売もしております。

全国通信販売
北海道から沖縄まで

★送料はご注文の際にお問い合わせ下さい。
★掲載の商品は、すべて新品、保証書付きです。
★掲載の商品は充分用意してありますが、ご注文の際は、在庫の確認の上、現金書留または、銀行振込でお申し込み下さい。全商品クレジットでも扱っております。
★お申し込みの際は必ず電話番号を明記して下さい。
★商品、品切れの節はご容赦下さい。

富士銀行八王子支店 (普)1752505

●本誌発売時には上記価格よりさらにお求めやすい価格に変更されている場合があります。●この広告の商品にはすべて送料・消費税は含まれておりません。

R&Rメディア オープンセール

↓NOW PLEASE CALL↓

☎03-3777-7335

パソコンショップ

R&Rメディア

特価セット

CZ-623C-TN	定価	¥498,000
CZ-606D-TN	定価	¥ 79,800
合計		¥577,800
R&R提供価格		¥378,000

CZ-604C-TN	定価	¥348,000
CZ-606D-TN	定価	¥ 79,800
合計		¥427,800
R&R提供価格		¥298,000

XVI お買い得セット

CZ-634C-TN	定価	¥368,000
CZ-606D-TN	定価	¥ 79,800
合計		¥447,800
R&R提供価格		¥3?0,000

ミュージックコンピュータセット

CZ-634C-TN	定価	¥368,000
CZ-606D-TN	定価	¥ 79,800
SX-68M	定価	¥ 19,800
CM-32L	定価	¥ 69,000
MA-12C(2台)	定価	¥ 28,000
MUSIC STUDIO PRO-68K V2.0	定価	¥ 28,000
合計		¥592,600
R&R提供価格		¥450,000

	定価	R&R提供価格
▼プリンタ(ケーブル付)		
CZ-8PC5	¥ 96,800	¥ 77,000
CZ-8PG1	¥130,000	¥104,000
IO-735XB	¥248,000	¥173,000
▼増設メモリ		
CZ-6BE1	¥ 35,000	¥ 27,800
PIO-6BE1A	¥ 25,000	¥ 19,800
PIO-6BE2	¥ 50,000	¥ 39,800
PIO-6BE4	¥ 88,000	¥ 69,800
CZ-6BE1B	¥ 28,000	¥ 21,800
CZ-6BE2A	¥ 59,800	¥ 47,800
▼その他のオプション		
CZ-6BS1	¥ 29,800	¥ 23,800
CZ-6VT1	¥ 69,800	¥ 56,000
CZ-6BM1	¥ 26,800	¥ 21,500
CZ-6BV1	¥ 21,000	¥ 16,900
CZ-8NS1	¥188,000	¥150,000
▼ソフトウェア		
SX-WINDOW V1.1	¥ 9,800	¥ 7,840
Easy Paint SX-68K	¥ 12,800	¥ 10,240
Multiword PRO-68K	¥ 32,000	¥ 25,600
Teleportion PRO-68K	¥ 22,800	¥ 18,240

ただ今、X68000のセットをお買い上げ頂きますと
「V'BALL」「熱血高校サッカー編」「ダウンタウン熱血物語」
のいずれか1本をプレゼント!

グラフィックセットコンピュータ

CZ-634C-TN	定価	¥368,000
CZ-606D-TN	定価	¥ 79,800
CZ-8PC5	定価	¥ 96,800
Z's STAFF PRO-68K V2.0	定価	¥ 58,000
合計		¥602,600
R&R提供価格		¥460,000

コンピュータミュージック

CM-32L	定価	¥ 69,000
SX-68M	定価	¥ 19,800
MUSIC STUDIO PRO-68K V2.0	定価	¥ 28,000
合計		¥116,800
R&R提供価格		¥ 97,000

CM-64	定価	¥129,000
SX-68M	定価	¥ 19,800
MUSIC STUDIO PRO-68K V2.0	定価	¥ 28,000
合計		¥176,800
R&R提供価格		¥149,000

SC-55	定価	¥ 69,000
SX-68M	定価	¥ 19,800
MUSIC STUDIO PRO-68K V2.0	定価	¥ 28,000
合計		¥116,800
R&R提供価格		¥ 97,800

※掲載商品の価格は、全て消費税別です。

通信販売ご利用にあたって

① 商品は、電話またはFAX(お客様の電話番号をお忘れなく)でご注文下さい。
② お支払は銀行振込でお願いいたします。入金確認後の発送となります。
③ ローンも扱っておりますので、ご相談下さい。
ローンは電話・FAXでのご注文後、申込用紙をお送りしますので、
必要事項に記入・捺印の上ご返送下さい。手続きしだい商品を発送します。
④ 送料・消費税は別途いただきますので、ご了承下さい。
⑤ 掲載分以外の商品も扱っておりますので、ご相談下さい。
振込先:富士銀行 西大井支店(普)1358191 アール・アンド・アール・メディア(株)

正社員を募集しています。

パソコンが好きで、夢と野心にあふれた人がいいですね。……お待ちしています。

お問合せ・お申込みはTEL,FAXにてお願い致します。

アール・アンド・アール・メディア株式会社
〒140 東京都品川区西大井6-10-10
TEL.03-3777-7335
FAX.03-3777-6448

●取扱い商品 NEC・富士通・エプソン・シャープ(メーカー保証付) ソフト、各種サプライ用品

ACCESS

X1エミュレータ

好評発売中

定価¥9,800

X1エミュレータはX68000上でX1シリーズのアプリケーションを実行するためのソフトエミュレータです。X1のアプリケーションを完全にソフトウェアのみでエミュレートしているため、X1上での実行速度と比較して、平均3～5倍程度おそくなりますが、X68000のマシン上に実現した仮想X1マシンを楽しめます。また、X1とX68000の相互間でファイルを転送するためのユーティリティと専用ケーブルが付属しますので、X1上で作り上げたソフトの資産をX68000上に移行することも簡単にできます。

X1エミュレータの機能

- X1エミュレータはX1に相当する機能をエミュレート。この仮想コンピュータには最大4つのドライブが仮想的に接続。
- X1エミュレータからみたドライブはHuman68kのドライブ上にあるファイルで仮想的に実現。このファイルはX1用の5″2Dディスクのイメージをファイル転送ユーティリティでまるごと転送したもの。
- X1エミュレータで仮想的に実現したX1は仮想ドライブから起動。このため仮想ドライブ用ファイルには、X1を立ち上げるために必要なHuBASICやCP/Mなどのシステムプログラムが必要。
- X1エミュレータでは、X1の持つVRAMを含むメモリイメージやZ80CPUを仮想的にソフトウェアで実現。

ファイル転送ユーティリティ

ディスク転送

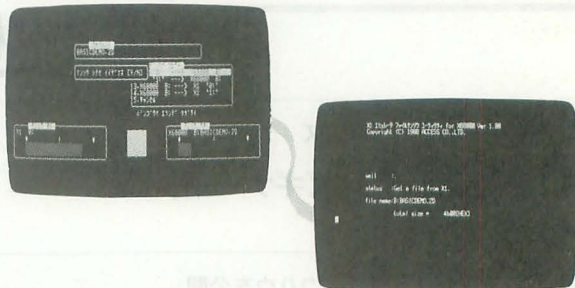
X1ディスク⟷X68000 Human68k(5″2Dディスクイメージファイル)

- X1エミュレータではHuman68k上のディスクイメージファイルを仮想ドライブとして使用。

ファイル転送

X1 BASIC:CP/M⟷X68000 Human68k

- X1で作ったプログラム&データをX68000上で使用。

※付属の専用ケーブルをX1とX68000に接続してファイルを転送します。

X1エミュレータ Q&A

Q. ファイル転送のために別途RS-232Cケーブルを買わないといけないのですか?
A. 専用のケーブルが付属しますのでその必要はありません。

Q. X1BASICのプログラムをX68000上のX-BASICで使えますか?
A. 通常のセーブではコードが違うので使用できませんが、アスキーセーブしたファイルであればX-BASIC上でそのままロード可能です。

Q. TurboBASICで作成した住所録などの漢字を含んだデータがあるのですがX68000上にファイル転送できますか?
A. X1TurboもX68000も漢字はシフトJISコードなのでファイルの転送は可能です。ただし、漢字ROMを必要とするものはサポートしていません。

Q. Turbo用のソフトは動きますか?
A. X1用のみでTurbo専用のソフトは動きません。

Q. ゲームは動きますか?
A. 純粋にBASICでかかれたものは動きますが、プロテクトがかかったものや直接ハードをアクセスするような市販のゲームは動きません。

*タイミング等ハードウェアに依存するようなソフトは、原理上実行できない、もしくは正常に動作しない場合がありますのでご注意ください。
*一部サポートしていない機能があります。

X1エミュレータ通信販売 購入希望として住所、氏名、電話番号をお知らせください。注文書をお送り致します。

*この商品価格には消費税は含まれておりません。
*CP/Mはデジタルリサーチ社の商標です。
文中のソフトウェアは各社の商標です。
*製品の仕様、名称は予告なく変更する場合もございますのであらかじめご了承ください。

有限会社アクセス 〒101 東京都千代田区神田神保町1-64 神保町協和ビル7F
☎03(3233)0200(代) FAX.03(3291)7019

資料請求券 oh!X 9月号

パソコン/ワープロ通信ネットワークサービス

J&P HOT LINE

J＆P HOTLINEは
私の社会への窓口です。

木田 茂夫さん 31歳

(JH082511 KID'S)

コピーライター

趣味が高じて仕事になった……というお手本がこの木田さん。１つのことに取り組むと昼も夜も忘れ、周囲の怪訝な目にも気付かず(!?)、とにかく納得するまで極めてしまうというタイプ。コンピュータとの出会いもご他聞に漏れず。で、今ではすっかりテクニカルライターとして活躍中。

電子メールで時間を短縮する、「一匹狼共同体」隊長。

木田さんは、フリーランスのコピーライター。とくにコンピュータ関係の仕事に的をしぼって活躍中です。広告・カタログ・企画と幅広い仕事に加えて、取材旅行にも出なければならない木田さんにとって、ノートパソコンとJ&P HOTLINEの電子メール機能はなくてはならない強い味方。締切が迫っても新幹線の中で原稿を書き上げ、ホテルから電子メールで送れば、手持ち時間は倍以上に活用できるとか。

アクティブな木田さんはJ&P HOTLINEの中でも、ライターのためのSIG「書くネット！」のSIGOPとして活動中。ブラインドタッチの練習ソフトを開発したり、演劇集団やプランナーなど他方面の「ライター」と交流を深めて、「書く情報交換」の可能性を追求しています。

また仕事をベースにした、オフラインの仲間とオンラインの仲間をつなぐネットワークも計画。孤立しがちなフリーランスライターにとって、広い視野と豊かなネットワークを得られるJ&P HOTLINEはなくてはならない存在です。

ネットワーキングだけでなく、J＆P HOT LINEは木田さんの巨大な情報ソース。

- ★豊富なメニューのBBSで、社会の動きをすばやくキャッチ。
- ★書くためのツールはX-MODEMで取り込んだフリーソフトを最大限に活用。書く効率は数倍に。
- ★いそがしい時は新刊書籍情報で読みたい本をチェックする。
- ★ソフト関連の記事を書く時は、新作ソフト情報・ソフトウェア情報でスペックを確認。
- ★趣味の料理のネタ探しに電子レンジ教室もときおりのぞく。

J&P HOTLINEへのご入会はスタータキットで。

お求めは、下記のお店へ。又は現金書留にて、¥3,000＋¥90(消費税3%)＝¥3,090を事務局までお送り下さい。
すぐにスタータキットをお送りします。

お問い合わせは 〒556 大阪市浪速区日本橋西1-6-5 上新電機株式会社
J&P HOTLINE事務局宛 TEL.(06)632-2521

スタータキットのお求めはJ&P各店でどうぞ。

店名	住所	電話
渋谷店	東京都渋谷区道玄坂2丁目28番4号	☎(03)3496-4141
町田店	東京都町田市森野1丁目39番16号	☎(0427)23-1313
八王子店	東京都八王子市旭町1番1号八王子そごう7F	☎(0426)26-4141
立川店	東京都立川市幸町4－39－1	☎(0425)36-4141
本厚木店	厚木市中町3－4－3	☎(0462)25-1548
富山店	富山市桜町2－1－10	☎(0764)32-3133
金沢店	金沢市入江2－63	☎(0762)91-1130
寺地店	金沢市寺地2－3	☎(0762)47-2524
大須店	名古屋市中区大須4丁目2-48	☎(052)262-1141
テクノランド	大阪市浪速区日本橋5丁目6番7号	☎(06) 634-1211
メディアランド	大阪市浪速区日本橋5丁目8番26号	☎(06) 634-1511
コスモランド	大阪市浪速区難波中2丁目1番17号	☎(06) 634-3111
U. S. LAND	大阪市浪速区日本橋4丁目9番15号	☎(06) 634-1411
ビジネスランド	大阪市北区梅田1-1-3大阪駅前第3ビルB2	☎(06) 348-1881
梅田店	大阪市北区小松原町1－10	☎(06) 362-1141
高槻店	高槻市高槻町11番16号	☎(0726)85-1212
くずは店	枚方市楠葉花園町15番2号	☎(0720)56-8181
千里中央店	豊中市新千里東町1-3 SENCHU PAL 2番街4F	☎(06) 834-4141
摂津富田店	高槻市大畑町24－10	☎(0726)93-7521
寝屋川店	寝屋川市緑町4－20	☎(0720)34-1166
藤井寺店	藤井寺市岡2丁目1番33号	☎(0729)38-2111
岸和田店	岸和田市土生町2451－3	☎(0724)37-1021
さんのみや1ばん館	神戸市中央区八幡通3-2-16	☎(078)231-2111
西宮店	兵庫県西宮市河原町5－11	☎(0798)71-1171
伊丹店	伊丹市昆陽池1－63	☎(0727)77-5101
姫路店	姫路市東延末1丁目1番住友生命姫路南ビル1F	☎(0792)22-1221
京都寺町店	京都市下京区寺町通仏光寺下ル恵比須之町549	☎(075)341-3571
京都近鉄店	京都市下京区烏丸通七条下ル東塩小路町702	☎(075)341-5769
和歌山店	和歌山市元寺町4丁目4番地	☎(0734)28-1441
奈良1ばん館	奈良市三条町478－1	☎(0742)27-1111
郡山インター店	大和郡山市横田693－1	☎(07435)9-2221
熊本店	熊本市手取本町4－12	☎(096)359-7800

T4910217909606 雑誌 02179-9